

**TU Braunschweig - Niedersächsisches
Forschungszentrum für Luftfahrt**

Berichte aus der Luft- und Raumfahrttechnik

Forschungsbericht 2017-24

Sliding-Mode-Lernverfahren für neuronale Netzwerke in adaptiven Regelungssystemen

Philipp Schnetter

TU Braunschweig
Institut für Flugführung

Diese Arbeit erscheint gleichzeitig als von der Fakultät für Maschinenbau der Technischen Universität Carolo-Wilhelmina zu Braunschweig zur Erlangung des akademischen Grades eines Doktor-Ingenieurs genehmigte Dissertation.

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Philipp Schnetter

Sliding-Mode-Lernverfahren für neuronale Netzwerke in adaptiven Regelungssystemen

©2017

ISBN 978-3-928628-93-8

Diese Arbeit erscheint gleichzeitig als von der Fakultät für Maschinenbau der Technischen Universität Carolo-Wilhelmina zu Braunschweig zur Erlangung des akademischen Grades eines Doktor-Ingenieurs genehmigte Dissertation.

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten. Mit Genehmigung des Autors ist es gestattet, diesen Bericht ganz oder teilweise zu vervielfältigen.

Herausgeber der NFL Forschungsberichte:

TU Braunschweig - Niedersächsisches Forschungszentrum für Luftfahrt

Hermann-Blenk-Straße 27

38108 Braunschweig

Tel: 0531-391-9822

Fax: 0531-391-9804

Mail: nfl@tu-braunschweig.de

Internet: www.nfl.tu-braunschweig.de

Copyright Titelbild: Philipp Schnetter

Sliding-Mode-Lernverfahren für neuronale Netzwerke in adaptiven Regelungssystemen

Von der Fakultät für Maschinenbau
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung der Würde

eines Doktor-Ingenieurs (Dr.-Ing.)

genehmigte Dissertation

von: Philipp Schnetter
aus: Kassel

eingereicht am: 16.03.2017
mündliche Prüfung am: 27.10.2017

Gutachter: Prof. Dr.-Ing. Peter Vörsmann
Prof. Dr.-Ing. Florian Holzapfel
Prof. Dr.-Ing. habil. Joachim K. Axmann

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter an dem Institut für Luft- und Raumfahrtsysteme sowie dem Institut für Flugführung der Technischen Universität Braunschweig in den Jahren 2011 bis 2017.

Ich bedanke mich bei Herrn Prof. Dr.-Ing. Peter Vörsmann für die Betreuung und sein Vertrauen den Schwerpunkt meiner Forschungstätigkeiten eigenverantwortlich wählen zu können. Bei Prof. Dr.-Ing. habil. Joachim Axmann bedanke ich mich für die stetige Unterstützung und den kontinuierlichen Austausch zu aktuellen Forschungsthemen. Ebenso danke ich Prof. Dr.-Ing.- Florian Holzapfel für die Übernahme des Korreferats und die allzeit spannenden inhaltlichen Diskussionen sowie Prof. Dr.-Ing. Peter Hecker für die Führung des Prüfungsvorsitzes und die Möglichkeit zum Abschluss meiner Forschungstätigkeiten an seinem Institut.

Bei meinen Kollegen möchte ich mich für Ihre Hilfsbereitschaft, die gemeinsamen fachlichen Gespräche sowie die stets freundschaftliche Arbeitsatmosphäre bedanken. Ein besonderer Dank gebührt dabei den Kollegen aus der Konzernforschung von Volkswagen, allen voran Jonas Kaste für die investierte Zeit auf dem Weg, die entwickelten Methoden auf die Straße zu bekommen. Besondere Erwähnung verdient Herr Dr.-Ing. Thomas Krüger, dessen Vorarbeit, die unzähligen gemeinsamen Diskussionen und kritischen Anregungen maßgeblich zum Gelingen dieser Arbeit beigetragen haben.

Mein herzlicher Dank gilt außerdem meinen Freunden und meiner Familie für die nie endende Unterstützung und ihre stets anspornende Neugier zum Stand meiner Forschung. Insbesondere meinen Eltern Renate und Michael danke ich für das geduldige und wertvolle Lektorat in der Endphase der Dissertation.

Ganz besonderer Dank gilt meiner Partnerin Monique, deren Geduld, Aufopferung und unerschütterlicher Glaube an die Richtigkeit und den Erfolg meines Tuns auf unschätzbare Weise zur Fertigstellung dieser Arbeit beigetragen haben.

Übersicht

Die vorliegende Arbeit stellt einen Beitrag zur Nutzung von Online-trainierten neuronalen Netzwerken in sicherheitskritischen Regelungssystemen dar. Indem der Trainingsfehler und dessen Ableitung durch die vorgestellten Verfahren in eine stabile Ruhelage gezwungen werden, ist es möglich das Konvergenzverhalten des Netzwerks gezielt zu beeinflussen. Als Resultat dessen, kann das oftmals als Black-Box empfundene Netzwerktraining gegenüber klassischen Methoden transparenter gestaltet werden.

Grundlage der vorgestellten Lernverfahren ist die Übertragung ausgewählter Sliding-Mode-Control-Konzepte erster und zweiter Ordnung auf das Training neuronaler Netzwerke. Durch die Verwendung der diesen Ansätzen zu Grunde liegenden Stabilitätsbedingungen, kann für die dargestellten Trainingsverfahren eine dynamische Berechnung der Lernrate umgesetzt werden. Die auf diese Weise realisierbare Überführung der Netzwerkzustände in den asymptotisch stabilen Gleitzustand wird anhand ausgewählter Trainingsbeispiele demonstriert und hinsichtlich des Konvergenzverhaltens in Abhängigkeit eingeführter Stabilitätsparameter untersucht.

Anwendung im Kontext der adaptiven Regelung sicherheitskritischer Systeme finden die dargestellten Trainingsverfahren als Teil einer nichtlinearen dynamischen Inversionsregelung. Zur Darstellung der generischen Natur der entwickelten Verfahren werden dabei sowohl ein Flug- als auch ein Fahrzeugregler mit Erweiterung um neuronale Netzwerke umgesetzt und in nichtlinearen Simulationen untersucht. Dabei unterstreichen die Ergebnisse sowohl die im Vergleich zu etablierten Ansätzen erhöhte Konvergenzgeschwindigkeit, als auch die gestiegene Robustheit gegenüber Parameterunsicherheiten und Systemschäden. In besondere Weise kann die vereinfachte Anwendbarkeit von neuronalen Netzwerken mit Verwendung der entwickelten Lernverfahren, durch die direkte Übertragbarkeit der Trainingsparameter zwischen den beiden vorgestellten Regelungsanwendungen, gezeigt werden.

Darüber hinaus demonstrieren praktische Fahrversuche die Relevanz von adaptiven Regelungssystemen für den Ausgleich auftretender Systemfehler. Dabei wird durch die Untersuchung des Sliding-Mode-Lernverfahrens sowie dem Vergleich des Inversionsreglers mit einem nicht lernfähigen Regelungskonzept der Vorteil von schnellen und robust trainierten neuronalen Netzwerken unterstrichen.

Abstract

The present thesis makes a contribution to the use of *online*-trained neural networks in safety critical control systems. By forcing the training error and its derivation into the stable point of origin, it is possible to directly influence the convergence behavior of the network. As a result, the network training often perceived as *black box* can be made more transparent compared to classical methods.

The basis of the presented learning methods is the transfer of selected first and second order sliding mode control concepts to the training of neural networks. Through the use of the intrinsic stability conditions, a dynamic calculation of the learning rate can be implemented for the presented training methods. The transition of the network states into the asymptotically stable sliding mode is demonstrated by means of selected training examples and examined with respect to the convergence behavior as a function of the introduced stability parameters.

Applied in the context of the adaptive control of safety-critical systems, the presented training methods become part of a non-linear dynamic inversion controller. To illustrate the generic nature of the developed methods, both a flight controller and a vehicle controller with extension of neural networks are implemented and tested in non-linear simulations. The results underline the increased speed of convergence as compared to established approaches as well as the increased robustness against parameter uncertainties and system failures. In particular, the simplified applicability of neural networks using the developed learning methods can be demonstrated. Due to the direct transferability of the training parameters between the two control applications presented, a time-consuming empirical determination of the learning rate can be avoided.

In addition, practical driving tests demonstrate the relevance of adaptive control systems for the compensation of system errors. The advantage of fast and robustly trained neural networks is underlined by the study of the sliding mode learning method and the comparison of the inversion controller with a non-adaptive control concept.

Inhaltsverzeichnis

Abbildungsverzeichnis	X
Tabellenverzeichnis	XV
Formelzeichen	XVII
1 Einleitung	1
1.1 Stand der Technik	3
1.2 Gegenstand der Arbeit	5
1.3 Gliederung der Arbeit	6
2 Grundlagen neuronaler Netzwerke	7
2.1 Neuronale Netze im Feld der künstlichen Intelligenz	8
2.2 Technische Umsetzung eines biologischen Vorbilds	10
2.2.1 Das künstliche Neuron	11
2.2.2 Netzwerktopologien	12
2.3 Lernverfahren	18
2.3.1 Backpropagation-Algorithmus	19
2.3.2 Herausforderungen der Nutzung Online-trainierter neuronaler Netzwerke in Regelungsanwendungen	25
3 Strukturvariable Regelung	27
3.1 Gleitzustandsregelung	30
3.1.1 Verhalten im Gleitzustand	32
3.1.2 Erreichbarkeit und Existenz des Gleitzustands	34
3.1.3 Ideales und reales Schalten	36
3.2 Gleitzustandsverfahren höherer Ordnung	38
3.2.1 Twisting-Algorithmus	40
3.2.2 Erreichbarkeits- und Existenzbedingung des Twisting-Algorithmus	43
4 Gleitzustandstrainingsverfahren für neuronale Netzwerke	45
4.1 Regelung des Netzwerkfehlers	46
4.2 Definition der Schaltfunktion	47
4.3 Gleitzustandstraining erster Ordnung	48

4.3.1	Beschreibung des Stellgesetzes	50
4.3.2	Berechnung der Lernrate	51
4.3.3	Untersuchung des Gleitzustandstrainings erster Ordnung	58
4.4	Gleitzustandstraining zweiter Ordnung	74
4.4.1	Berechnung der Lernrate mit erweiterter Kontraktionsbedingung . . .	75
4.4.2	Untersuchung des Einflusses von $\tau_{\dot{s}}$ auf das Netzwerktraining	79
4.4.3	Untersuchung der gegenseitigen Beeinflussung von τ , $\tau_{\dot{s}}$ und λ	85
5	Regelungskonzept	91
5.1	Nichtlineare dynamische Inversion	93
5.1.1	Grundlegendes Prinzip der dynamischen Inversion	93
5.1.2	Linearisierende Zustandsrückführung	98
5.1.3	Referenzgrößen und Inversionsfehler	100
5.1.4	Linearregler der Fehlerdynamik	102
5.1.5	Pseudo-Control Hedging	104
5.2	Erweiterung durch neuronale Netzwerke	107
5.2.1	Fehlerdynamik des erweiterten Systems	109
5.2.2	Beschränktheitsbetrachtung des adaptiven Reglers	110
6	Simulationsumgebung und Reglerumsetzung	115
6.1	Umsetzung als Flugregler	116
6.1.1	Simulationsumgebung	116
6.1.2	Reglerumsetzung	120
6.2	Umsetzung als Fahrzeugregler	130
6.2.1	Simulationsumgebung	130
6.2.2	Reglerumsetzung	132
7	Darstellung ausgewählter Ergebnisse	139
7.1	Flugsimulation	139
7.1.1	Nominalfall mit leichten Parameterunsicherheiten	140
7.1.2	Netzwerkverhalten bei sprunghaftem Inversionsfehler	143
7.1.3	Erhebliche Parameterunsicherheiten mit Schwerpunktverschiebung .	147
7.2	Fahrsimulation	151
7.2.1	Langzeituntersuchung mit leichten Parameterunsicherheiten	151
7.2.2	Netzwerkverhalten bei Reifenplatzer	155
7.3	Fahrversuch	156
7.3.1	Fehlerausgleich auf einfachem Rundkurs	157
7.3.2	Fehlerausgleich auf anspruchsvollem Rundkurs	161
8	Zusammenfassung und Ausblick	165
8.1	Zusammenfassung	165
8.2	Ausblick und künftige Fragestellungen	170

Literaturverzeichnis	173
A Abbildungen zu den Gleitzustandsverfahren	185
B Erweiterung der Beschränktheitsuntersuchung	189
C Weitere Ergebnisse	191
C.1 Flugsimulation	191
C.2 Fahrsimulation	192
C.3 Fahrversuch	194

Abbildungsverzeichnis

2.1	Darstellung eines vernetzten künstlichen Neurons mit drei Ein- und Ausgabegrößen.	11
2.2	Schematische Darstellung verschiedener Transferfunktionen	12
2.3	Feedforward-Netzwerk mit Verbindungsmatrix	13
2.4	Shortcut-Netzwerk mit Verbindungsmatrix	14
2.5	Direct-Recurrent-Netzwerk mit Verbindungsmatrix	15
2.6	Indirect-Recurrent-Netzwerk mit Verbindungsmatrix	15
2.7	Darstellung eines vorwärtsgerichteten Netzes mit linearen Identitätsfunktionen in der Ein- und Ausgabeschicht, sowie sigmoider Funktion in der einen verdeckten Schicht.	16
2.8	Beispielhafte Fehlerfunktion für die Berechnung eines Gradienten zur Findung des Minimums durch den Backpropagation-Algorithmus.	20
2.9	Beispielnetzwerk mit zwei Ausgabeneuronen zur Veranschaulichung der Jacobi-Matrix Einträge	21
3.1	Abstrakte Darstellung eines strukturvariablen Systems.	27
3.2	Darstellung zweier grenzstabiler Strukturen in der Phasenebene	28
3.3	Darstellung der kombinierten Systemdynamik von Struktur 1 und 2.	29
3.4	Strukturvariable Regelung im Gleitzustand.	30
3.5	Beispielhafte Darstellung der Zustandstrajektorien im Phasenraum für (a) unterschiedliche Schaltfunktionen und (b) unterschiedliche Systemeigenschaften	32
3.6	Geometrische Lösung der Systemdynamik f_0 im Gleitzustand nach Filipov	33
3.7	Darstellung der Zustandstrajektorien des Twisting-Algorithmus im Phasenraum mit Schalten an verschiedenen Schaltregimen.	41
3.8	Abstrakte Darstellung der dreidimensionalen Phasentrajektorie einer Gleitzustandsregelung zweiter Ordnung mit diskontinuierlichem Verhalten in \dot{S}	42
4.1	Das Netzwerktraining als geschlossener Regelkreis.	46
4.2	Lösung der Schaltfunktion für $S = 0$ bei (a) $\lambda > 0$ und (b) $\lambda < 0$	48
4.3	Phasenportrait eines instabilen diskreten Systems bei Verwendung einer zu großen Schrittweite T_s in Kombination mit einer kontinuierlichen Erreichbarkeitsbedingung.	49

4.4	Lösungsintervall der Lernratenberechnung für die diskrete Erreichbarkeits- und Existenzbedingung $ S_{t+1} < S_t $	55
4.5	Blockschaltbild des Gleitzustandstrainings erster Ordnung für ein Netzwerk mit einer Ausgangsgröße	56
4.6	Darstellung des Intervallparameters τ zur Angabe einer Position im Lernratenintervall	57
4.7	Zielgröße des Netzwerktrainings mit zweifachem Sprung	59
4.8	Phasenportrait eines mit dem Gleitzustandsverfahren erster Ordnung trainierten neuronalen Netzwerks mit zwei Sprüngen der Zielgröße	60
4.9	Netzausgaben und Schaltfunktionswerte für verschiedene Intervallparameter bei einem Sprung der Zielgröße	61
4.10	Lernraten und Schaltfunktionsvorzeichen für verschiedene Intervallparameter bei einem Sprung der Zielgröße	62
4.11	Netzausgaben und σ^* -korrigierte Lernraten für verschiedene Intervallparameter bei einem Sprung der Zielgröße	65
4.12	Komplexer Zielgrößenverlauf des Netzwerktrainings	66
4.13	Vergleich der Netzerkzeugausgaben für einen komplexen Zielgrößenverlauf mit verschiedenen Intervallparametern und $\lambda = 50$ sowie σ^* -Korrektur	66
4.14	Vergleich der berechneten Lernraten mit $\lambda = 50$ sowie σ^* -Korrektur	68
4.15	Vergleich der Netzerkzeugausgaben für einen komplexen Zielgrößenverlauf mit Erreichung und Überschreitung der Grenzen des Lernratenintervalls und $\lambda = 50$ ohne σ^* -Korrektur	68
4.16	MSE Vergleich für verschiedene λ in Abhängigkeit des Intervallparameters τ für das Netzwerktraining mit σ^* -Korrektur	69
4.17	Ausschnitt des MSE Vergleichs für verschiedene λ in Abhängigkeit des Intervallparameters τ für das Netzwerktraining mit σ^* -Korrektur	70
4.18	Oberflächendarstellung des mittleren quadratischen Fehlers für verschiedene λ in Abhängigkeit des Intervallparameters τ für das Netzwerktraining mit σ^* -Korrektur	71
4.19	Vergleich der Netzerkzeugausgaben für zwei verschiedene Kombinationen von λ und τ	72
4.20	Vergleich der zwei Parameterkombinationen des Gleitzustandstrainings erster Ordnung mit dem Backpropagation-Algorithmus	73
4.21	Phasenportraits der unterschiedlichen Lernverfahren für das komplexe Trainingsszenario	74
4.22	Blockschaltbild des Gleitzustandstrainings zweiter Ordnung für ein Netzwerk mit einer Ausgangsgröße	79
4.23	Phasenportrait eines mit dem Gleitzustandsverfahren zweiter Ordnung trainierten neuronalen Netzwerks mit zwei Sprüngen der Zielgröße	80
4.24	Phasenportrait eines mit dem Gleitzustandsverfahren zweiter Ordnung trainierten neuronalen Netzwerks mit einem Sprung der Zielgröße	81

4.25	Netzausgaben und Schaltfunktionswerte für verschiedene Intervallparameter bei einem Sprung der Zielgröße	82
4.26	Phasenportrait eines mit dem Gleitzustandsverfahren zweiter Ordnung trainierten neuronalen Netzwerks mit Begrenzung der kombinierten Lernrate durch das Verfahrens erster Ordnung	83
4.27	Netzausgaben und Schaltfunktionswerte mit Begrenzung der kombinierten Lernrate durch das Verfahrens erster Ordnung	84
4.28	Kombinierte Lernrate für verschiedene Intervallparameter bei einem Sprung der Zielgröße	84
4.29	MSE Vergleich in Abhängigkeit von τ , τ_s und λ für $\lambda = [50...550]$	86
4.30	MSE Vergleich in Abhängigkeit von τ und τ_s für $\lambda = 800$ mit Referenzwert des Verfahrens erster Ordnung	88
4.31	Vergleich der Netzausgaben für drei verschiedene Kombinationen von τ und τ_s mit SMC-FO1	89
5.1	Abstrakte Darstellung des transformierten Systems mit linearisierender Zustandsrückführung	100
5.2	Lineares Referenzmodell zweiter Ordnung	101
5.3	Struktur eines Inversionsreglers für ein Eingrößensystem	104
5.4	Referenzmodell zweiter Ordnung mit PCH	106
5.5	Struktur eines Inversionsreglers für ein Eingrößensystem mit PCH	106
5.6	Struktur eines Inversionsreglers für ein Eingrößensystem mit PCH und neuronalem Netzwerk	108
6.1	Blockschaltbild der nichtlinearen Simulationsumgebung	116
6.2	Blockschaltbild der Flugdynamiksimulation	119
6.3	Kompensation der Bahnablage	121
6.4	Vereinfachtes Blockschaltbild des kaskadischen Reglers	122
6.5	Umsetzung der Inversionsregelung der Rotationsdynamik	124
6.6	Signalfluss der Fahrzeugdynamiksimulation	131
6.7	Skalierung der Fehlergröße e_r in Abhängigkeit der seitlichen Bahnablage d	134
6.8	Kräfte und Winkel an der Vorderachse des Fahrzeugs	135
6.9	Aufbau der Steuerallokation zur Lenkwinkelberechnung	136
7.1	Bahnvorgabe der Flugsimulation in Längs- und Seitenbewegung	140
7.2	Größen der Längsdynamikregelung mit leichten Parameterunsicherheiten und Windeinfluss	141
7.3	Größen der Seitendynamikregelung mit leichten Parameterunsicherheiten und Windeinfluss	142
7.4	Reaktion auf sprunghaften Inversionsfehler in der Längsbewegung	144
7.5	Ausgleich eines sprunghaften Inversionsfehlers bei unterschiedlichen Intervallparametern	145

7.6	Lernratenvergleich für den Ausgleich eines sprunghaften Inversionsfehlers bei unterschiedlichen Intervallparametern	146
7.7	Netzwerkausgabe und Änderung der statischen Stabilität im Schadensfall . .	147
7.8	Vergleich des Höhenfehlers der verschiedenen Lernverfahren im Schadensfall	148
7.9	Vergleich des Höhenfehlers der beiden Lernverfahren im Schadensfall mit zusätzlichem sprunghaftem Inversionsfehler	149
7.10	Rundkurs für die Untersuchung des Fahrzeugreglers	151
7.11	Fehlervergleich der Gierratenregelung für mehrere Runde mit SMC-FO . . .	152
7.12	Vergleich der Bahnabweichung für mehrere Runde mit SMC-FO	153
7.13	Frobeniusnorm für das Langzeittraining mit SMC-FO	153
7.14	Änderung der Reifenkräfte und -momente in Folge eines Reifenplatzens nach [142]	155
7.15	Vergleich der Bahnabweichung für unterschiedliche Lernverfahren mit Reifenplatzer bei Sekunde 25	156
7.16	Einfacher Rundkurs für die Untersuchung des Fahrzeugreglers im Fahrversuch	157
7.17	Skalierte Geschwindigkeitsvorgabe für die automatische Fahrt mit Fehleraufschaltung	158
7.18	Vergleich der Bahnabweichungen für die unterschiedlichen Geschwindigkeitsvorgaben mit Fehleraufschaltung	159
7.19	Vergleich der Pseudosteuergrößenanteile für die automatische Fahrt mit Fehleraufschaltung bei $V_{Skal} = 0,92$	159
7.20	Vergleich der Bahnabweichungen des Referenz- und des Inversionsreglers für die automatische Fahrt mit Fehleraufschaltung	160
7.21	Anspruchsvoller Rundkurs für die Untersuchung des Fahrzeugreglers im Fahrversuch	161
7.22	Vergleich der Bahnabweichungen für unterschiedliche Netzwerkparameter des Gleitzustandstrainings erster Ordnung	162
7.23	Vergleich der Frobeniusnorm für unterschiedliche Parameter des Gleitzustandstrainings erster Ordnung	162
7.24	Vergleich der Gierratenregelung für das Gleitzustandstrainings erster Ordnung mit $\lambda = 80$ und $\tau = 0,6$	163
A.1	Einträge des Eingabevektors der trainierten neuronalen Netzwerke für das komplexe Trainingsszenario	185
A.2	MSE Vergleich für verschiedene λ in Abhängigkeit des Intervallparameters τ für das Netzwerktraining ohne σ^* -Korrektur	186
A.3	Vergrößerter Ausschnitt des MSE Vergleichs für verschiedene λ in Abhängigkeit des Intervallparameters τ für das Netzwerktraining ohne σ^* -Korrektur .	186
A.4	Kombinierte Lernrate für verschiedene Intervallparameter bei einem Sprung der Zielgröße mit $\tau = 0,7$	187
A.5	MSE Vergleich in Abhängigkeit von τ , τ_s und λ für $\lambda = [650...950]$	187

C.1	Vergleich der Netzwerkausgabe für die Rollratenregelung mit Backpropagation- und Gleitzustandstraining im Schadensfall	191
C.2	Vergleich der Frobeniusnorm für ein Backpropagation- und Gleitzustandstraining im Schadensfall mit zusätzlich induziertem Inversionsfehler	192
C.3	Geschwindigkeitsvorgabe für die Langzeituntersuchung auf Rundkurs	192
C.4	Mittlerer quadratischer Fehler der Gierratenregelung über 90 Runden mit SMC-FO	193
C.5	Vergleich der Netzwerkausgabe für mehrere Runden mit SMC-FO	193
C.6	Vergleich der Quereschleunigungen für die unterschiedlichen Geschwindigkeitsvorgaben mit Fehleraufschaltung	194
C.7	Geschwindigkeitsvorgabe für den Fehlerausgleich auf anspruchsvollem Rundkurs	194

Tabellenverzeichnis

2.1	Definitionen von künstlicher Intelligenz nach [4]	9
4.1	Fallunterscheidung der zeitdiskreten Existenz- und Erreichbarkeitsbedingung	53
4.2	Konvergenzquotient für ausgewählte Intervallparameter	58
4.3	Fallunterscheidung des Ausgabefehlers	63
4.4	Fehlerauswertung bei unterschiedlichen Intervallparametern	67
4.5	Parameterkombination für das Gleitzustandstraining erster Ordnung (Sliding-Mode-Control-First-Order)	72
4.6	Fallunterscheidung der zeitdiskreten Kontraktionsbedingung für $\dot{S}_{t,i}$	75
4.7	Parameterkombinationen zur Erreichung des MSE von SMC-FO1	88
7.1	Fehlerauswertung im Schadensfall mit Inversionsfehler, Parameterunsicher- heiten und atmosphärischen Störungen	150
7.2	Fehlerauswertung für 60 minütigen fehlerfreien Nominalfall mit leichten Pa- rameterunsicherheiten	154

Formelzeichen

Lateinische Großbuchstaben

A	Auftriebskraft	[N]
$\mathbf{A}(\vec{x})$	Entkopplungsmatrix	[–]
\mathbf{A}_E	Systemmatrix der Fehlerdynamik	[...]
$\mathbf{A}(x)$	Systemmatrix der Zustandsraumdarstellung	[...]
$\mathbf{B}(x)$	Eingangsmatrix der Zustandsraumdarstellung	[...]
C	aerodynamischer Beiwert, aerodynamisches Derivat	[–]
\vec{D}, D	(vektorieller) Drehimpuls	$[\text{kg} \cdot \frac{\text{m}^2}{\text{s}}]$
D_i	Hauptabschnittsdeterminante i	[–]
\mathbf{E}	Einheitsmatrix	[–]
E	Quadratischer Fehler des Netzwerktrainings	[–]
\vec{F}, F	(vektorielle) Kraft	[N]
$F(x, u)$	Nichtlineare Funktion; linear in der Steuerung	[...]
$\mathbf{G}(x)$	Matrix zur Beschreibung des Einflusses der einzelnen Steuergrößen	[...]
$G(s)$	Übertragungsfunktion	[...]
\vec{G}, G	(vektorielle) Gewichtskraft	[N]
\mathbf{H}	Hurwitzmatrix	[...]
H	Flughöhe	[m]
\mathbf{J}	Jacobimatrix	[...]
\mathbf{K}	Verstärkungsmatrix	[–]
L	Rollmoment	[Nm]
$L_{ij}(x)$	Lie-Ableitung von $j(x)$ entlang des Vektorfeldes $i(x)$	[–]

M	Nickmoment	[Nm]
\vec{M}	Momentenvektor	[Nm]
\mathbf{M}_{ij}	Transformationsmatrix vom j-System in das i-System	[–]
N	Giermoment	[Nm]
O	Restterm der Taylorreihenentwicklung	[–]
\mathbf{P}_E	Positiv definite, symmetrische Matrix der linearen Ljapunow-Gleichung	[...]
Q	Querkraft	[N]
\mathbf{Q}	Verstärkungsmatrix des <i>reaching laws</i>	[–]
\mathbf{Q}_E	Positiv definite, symmetrische Matrix der linearen Ljapunow-Gleichung	[...]
\vec{R}^A	Vektor der aerodynamischen Kräfte	[N]
S	Bezugsflügelfläche	[m ²]
\vec{S}, S	(vektorielle) Schaltfunktion	[...]
\mathbf{T}	Trägheitstensor	[kg · $\frac{\text{m}^2}{\text{s}}$]
T	Zeitkonstante eines linearen Systems 1. Ordnung	[s]
T_s	Zeitschritt	[–]
\vec{V}, V	(vektorielle) Geschwindigkeit	[$\frac{\text{m}}{\text{s}}$]
V	Ljapunow-Funktion	[–]
W	Widerstand	[N]

Lateinische Kleinbuchstaben

a_i	Substitutionsparameter der Lernratenberechnung	[–]
a_y	Querbeschleunigung	[$\frac{\text{m}}{\text{s}^2}$]
$a(\vec{z})$	Substitutionsparameter der Zustandstransformation	[...]
$\vec{b}(x)$	Vektorfeld; unabhängig von Eingangsgrößen	[...]
b	Bezugsspannweite	[m]
b_i	Substitutionsparameter der Lernratenberechnung	[–]
\vec{b}_E	Eingangsvektor der Fehlerdynamik	[...]
$b(\vec{z})$	Substitutionsparameter der Zustandstransformation	[...]
c	Rückführkoeffizient der Fehlerdynamik	[–]

d	seitliche Bahnablage	[m]
\vec{e}, e	(vektorieller) Regelfehler	[...]
$\vec{f}(x)$	Vektorfeld; unabhängig von Eingangsgrößen	[...]
f_i^l	Übertragungsfunktion des i -ten Neurons in der l -ten Schicht	[—]
\vec{f}_s	Vektorfeld der externen Störungen	[—]
$\Delta \vec{f}$	Vektorfeld der internen Störungen	[—]
g	Erdbeschleunigung	$[\frac{\text{m}}{\text{s}^2}]$
$\vec{g}(x)$	Vektorfeld zur Beschreibung des Einflusses einer Steuergröße	[...]
$\vec{h}(x)$	Vektorfeld für die Beschreibung der Ausgangsgröße	[...]
k_{r_0}	Entwurfparameter des robustifizierenden Terms	[—]
k_{r_1}	Entwurfparameter des robustifizierenden Terms	[—]
k	Verstärkungsfaktor der Rückführung	[—]
l_μ	Bezugsflügeltiefe	[m]
l_v	Schwerpunktrücklage	[m]
m	Masse	[kg]
$n_j^{(l)}$	Neuroneneingang des j -ten Neurons in der l -ten Schicht	[—]
n	Lastvielfaches	[—]
n	Ordnungszahl	[—]
$o_j^{(l)}$	Neuronenausgabe des j -ten Neurons in der l -ten Schicht	[—]
p	Rollrate	$[\frac{\text{rad}}{\text{s}}]$
$\vec{p}(\vec{\xi}, \vec{\eta})$	Vektorfeld; von u abhängiger Teil der internen Dynamik	[...]
\vec{p}	Vektor unbekannter Systemparameter	[...]
q	Nickrate	$[\frac{\text{rad}}{\text{s}}]$
$\vec{q}(\vec{\xi}, \vec{\eta})$	Vektorfeld; von u unabhängiger Teil der internen Dynamik	[...]
\bar{q}	Staudruck	$[\frac{\text{kg}}{\text{m} \cdot \text{s}^2}]$
r	Gierrate	$[\frac{\text{rad}}{\text{s}}]$
\vec{r}, r	(vektorieller) relativer Grad	[—]

s	Laplace-Variable	[—]
\vec{s}	Positionsvektor	[—]
s_v	Halbspurweite der Vorderachse	[m]
t	Zeit	[s]
\vec{u}, u	(vektorielle) Steuergröße; Eingangsgröße eines dynamischen Systems	[...]
u	x-Komponente der Geschwindigkeit	$[\frac{m}{s}]$
v	y-Komponente der Geschwindigkeit	$[\frac{m}{s}]$
w	z-Komponente der Geschwindigkeit	$[\frac{m}{s}]$
\mathbf{w}	Gesamtgewichtsmatrix eines neuronalen Netzwerks	[—]
$\mathbf{w}^{(i)}$	Gewichtsmatrix der i -ten Schicht	[—]
$\Delta \mathbf{w}$	Gewichtsänderung eines neuronalen Netzwerks	[—]
w_{ij}	Gewichtsfaktor zwischen dem i -ten und j -ten Gewicht	[—]
x	x-Koordinate	[m]
\vec{x}	Zustandsvektor eines dynamischen Systems	[...]
\vec{x}	Eingangsvektor eines neuronalen Netzwerks	[...]
\vec{y}, y	(vektorielle) Ausgangsgröße eines dynamischen Systems	[...]
\vec{y}, y	(vektorielle) Ausgabe eines neuronalen Netzwerks	[...]
y	y-Koordinate	[m]
\vec{z}	Zustandsvektor des transformierten Systems	[...]
z	z-Koordinate	[m]
\mathbf{z}	Störterm der Netzlinearisierung	[—]

Griechische Buchstaben

α	Anstellwinkel	[rad]
α	Schräglaufwinkel	[rad]
$\vec{\alpha}(\vec{x})$	Koeffizientenvektor der linearisierenden Zustandsrückführung	[...]
β	Schiebewinkel	[rad]
β	Schwimmwinkel	[rad]
$\vec{\beta}(\vec{x})$	Koeffizientenvektor der linearisierenden Zustandsrückführung	[...]

χ	Bahnazimut	[rad]
$\vec{\chi}$	Vektor der Regelfehler und deren Ableitungen	[...]
$\vec{\Delta}, \Delta$	(vektorieller) Inversionsfehler; Modellfehler der Inversion	[...]
δ	Lenkwinkel	[rad]
δ_l	Lenkradwinkel	[rad]
δ_F	Schubhebelstellung	[—]
$\hat{\delta}_j^{(l)}$	Rückpropagierter Ausgang des j -ten Neurons in der l -ten Schicht	[—]
$\vec{\varepsilon}_A$	(vektorieller) Approximationsfehler	[—]
$\vec{\varepsilon}$	(vektorieller) Ausgabefehler eines neuronalen Netzes	[—]
Φ	Hängewinkel, Rolllage	[rad]
$\Phi(x)$	Funktion der Koordinatentransformation $z = \Phi(x)$	[...]
Γ	Konstante zur Bestimmung des Schräglaufwinkels	$[\frac{\text{N} \cdot \text{s}}{\text{rad} \cdot \text{m}}]$
γ	Bahnneigungswinkel	[rad]
η	Höhenruderausschlag	[rad]
$\vec{\eta}$	Teilzustandsvektor nach E/A-Linearisierung; beschreibt interne Dynamik des Systems	[...]
λ	Schaltfunktionsparameter in $S = \dot{x} + \lambda \cdot x$	[—]
λ	Parameter der e-Modifikation	[—]
μ_K	Bahnhängewinkel	[rad]
μ	Lernrate	[—]
\vec{v}	(vektorielle) Pseudosteuergröße	[...]
Θ	Nickwinkel	[rad]
ρ	Luftsdichte	$[\frac{\text{kg}}{\text{m}^3}]$
σ^*	Korrekturkoeffizient der Lernratenberechnung	[—]
σ_d	Faktor der e_r -Skalierung in der Rotationsdynamik	[—]
$\vec{\zeta}$	(vektorielle) gefilterte Fehlergröße	[—]
τ	Intervallparameter der Lernratenbestimmung	[—]
$\omega, \vec{\omega}$	Winkelgeschwindigkeit, Vektor der Winkelgeschwindigkeit	$[\frac{\text{rad}}{\text{s}}]$

ω_0	Ungedämpfte Eigenfrequenz	$[\frac{1}{s}]$
ξ	Seitenruderausschlag	[rad]
ξ	Teilzustandsvektor nach E/A-Linearisierung; beschreibt invertierte Dynamik des Systems	[...]
Ψ	Gierwinkel	[rad]
ζ	Seitenruderausschlag	[rad]
ζ	Dämpfung eines linearen Systems 2. Ordnung	[–]

Indizes

$\hat{}$	Modellgröße
$*$	Markierung der optimalen Gewichtsmatrix
\sim	Abweichung zwischen einer tatsächlichen und einer Modellgröße
0	Referenzzustand
A	aerodynamische Größe
a	Komponente im aerodynamischen Koordinatensystem
ad	Adaptives Element
<i>Bahn</i>	Größen der Bahndynamik
d	Auf die seitliche Bahnablage bezogene Größe
e	Anteil der linear geregelten Rückführung
eq	Markierung der äquivalenten Regelung
F	Markierung der Frobeniusnorm
f	Komponente im flugzeugfesten Koordinatensystem
g	Komponente im geodätischen Koordinatensystem; erdlotfest
h	Hedgesignal
I	Integraler Anteil
i	Neuron i
j	Neuron j
K	Bahngröße
k	Komponente im bahnfesten Koordinatensystem
k	kommandierte Größe

<i>Lage</i>	Größen der Lagedynamik
<i>lim</i>	Markierung einer Begrenzung
<i>l</i>	Bezogen auf das Rollmoment
<i>l</i>	Index der Netzwerkschichten
<i>l</i>	Markierung des Lenkradwinkels
<i>m</i>	Bezogen auf das Nickmoment
<i>n</i>	Bezogen auf das Giermoment
<i>P</i>	Proportionaler Anteil
<i>q</i>	Neuronenanzahl einer Schicht
<i>R</i>	Referenzsignal der dynamischen Inversion
<i>r</i>	Referenzwert
<i>r</i>	Auf die Gierdynamik bezogene Größe
<i>r</i>	Markierung des robustifizierenden Terms
<i>SR</i>	Bezogen auf eine Stabilitätsreserve
\dot{S}	Umschaltung an der ersten Schaltfunktionsableitung
<i>s</i>	Auf einen Zeitschritt bezogene Größe
<i>TW</i>	Durch den Antrieb erzeugte Kraft
<i>Tw</i>	Markierung des aus dem Twisting-Algorithmus abgeleiteten Terms
<i>t</i>	Markierung des aktuellen Zeitschritts
$t \pm 1$	Signal des vorangegangenen bzw. folgenden Zeitschritts
<i>V</i>	Auf die Vorderachse bezogene Größe
<i>X,x</i>	Komponente in Richtung; um x-Achse
<i>Y,y</i>	Komponente in Richtung; um y-Achse
<i>Z,z</i>	Komponente in Richtung; um z-Achse
δ	Bezogen auf den Lenkwinkel
η	Bezogen auf den Höhenruderausschlag
ω	Größe der Rotationsdynamik
ξ	Bezogen auf den Querruderausschlag
ζ	Bezogen auf den Seitenruderausschlag

Abkürzungen

BP	Backpropagation
FO	First-Order
ILS	Instrumentenlandesystem
KI	Künstliche Intelligenz
KNN	Künstliche neuronale Netze
MIMO	Multiple-Input-Multiple-Output; Mehrgrößensystem
MSE	Mean-Square-Error
PCH	Pseudo-Control-Hedging
SISO	Single-Input-Single-Output; Eingrößensystem
SMC	Sliding-Mode-Control
SO	Second-Order
UAS	Unmanned-Aircraft-System
UAV	Unmanned-Aircraft-Vehicle
V&V	Verifikation und Validation

Kapitel 1

Einleitung

Spätestens mit Einführung von Autopiloten für die automatische Kurs- und Höhenhaltung in der Fliegerei Mitte des 20. Jahrhunderts, haben sich Regelungssysteme für die Anwendung in sicherheitskritischen Systemen durchgesetzt und sind heute nicht mehr wegzudenken. Durch die Abgabe der eigentlichen Steuerung an den Flugregler können Piloten gerade auf langen Flugstrecken entlastet werden, so dass sie sich vollständig auf die Überwachung der Bordsysteme konzentrieren können. Zeitgleich kann durch den Einsatz der heutzutage mehrfach redundanten Autopilotensysteme das Risiko menschlichen Versagens in kritischen Situationen reduziert werden, so dass beispielsweise auf entsprechend ausgerüsteten Flughäfen auch im Fall geringer Sichtweiten vollautomatische Landungen möglich sind.

Angetrieben durch rasante Fortschritte auf den Gebieten der Elektronik, insbesondere der Halbleitertechnologie, konnten immer komplexere und leistungstärkere Computersysteme entwickelt werden, die ihrerseits die Entwicklung neuartiger Regelungsarchitekturen ermöglicht haben. So ist es heutzutage beispielsweise möglich, selbst hoch parallelisierte Algorithmen aus dem Forschungsfeld der künstlichen Intelligenz, wie beispielsweise vielschichtige neuronale Netzwerke mit hunderten Eingangsgrößen, in Echtzeit zu betreiben und somit für die Regelung nutzbar zu machen. Durch die hohe Verfügbarkeit solcher leistungsfähigen Regelungssystemen und ihrer technischen Umsetzung auf kleinstem Bauraum, haben in den Jahren seit der Jahrtausendwende automatisch fliegende unbemannte Fluggeräte (engl. Unmanned-Aerial-Vehicle - UAV) neben dem militärischen Einsatz auch verstärkt für zivile Anwendungen an Bedeutung gewonnen. Durch den Wegfall eines mitfliegenden menschlichen Piloten sind diese System zum einen in den eigentlichen Fluganwendungen deutlich weniger limitiert und zum anderen können sie durch die verwendeten Flugregelungssysteme nahezu von jedem Anwender ohne einen aufwändigen Schulungsprozess genutzt werden. Dies ist insbesondere für die im freien Handel vielfältig angebotenen Multicoptersysteme mit einer Abflugmasse von weniger als 2kg von Relevanz. Durch ihre einfache Handhabbarkeit und dem Angebot automatischer Flugmanövern, wie dem Verfolgen eines vorgegebenen Zieles, werden diese Systeme heutzutage für die unterschiedlichsten Anwendungen eingesetzt.

Im Januar 2017 wurde dabei ein Vorschlag zur Änderung des Betriebs von unbemannten Flugsystemen in Deutschland durch das Bundesministerium für Verkehr und digitale Infrastruktur (BMVI) vorgeschlagen¹. Dabei soll unter anderem umgesetzt werden, dass der gewerbliche Betrieb von UAVs außerhalb der Sichtlinie nun nicht mehr generell verboten ist und für Systeme oberhalb von 5kg von den jeweiligen Landesluftfahrtbehörden erlaubt werden kann. Diese Möglichkeit des Betreibens von unbemannten Flugsystemen ohne eine durch das menschliche Auge vorgegebene Reichweitenbeschränkung ermöglicht noch einmal einen deutlichen Hub in der Vielseitigkeit möglicher UAV-Anwendungen. Außerdem unterstreicht es einmal mehr die Notwendigkeit eines stabilen und leistungsfähigen Regelungssystems zum Ausgleich nicht absehbarer Störungen im automatischen Flugbetrieb ohne optische Kontrollmöglichkeit eines menschlichen Piloten.

Auch in der Automobilbranche hat die zur Verfügung stehende Regelungs- und Umfelderkennungstechnik zu der Forschung und Entwicklung an höher automatisierten Fahrzeugfunktionen geführt. So arbeiten verschiedene Automobilhersteller daran die Teilautomatisierung, in Form von dauerhaft zu überwachenden Assistenzfunktionen des Automatisierungsgrades 2, durch sogenannte hochautomatisierte Funktionen, bei denen der Fahrer das System nicht mehr überwachen muss, zu ersetzen.² Der Grad 3 beinhaltet beispielsweise eine automatische Fahrt auf der Autobahn, bei der der Fahrer erstmals nicht die Hände am Lenkrad haben muss, sich so auf Nebentätigkeiten konzentrieren darf und erst im Falle eine Übernahmeaufforderung die manuelle Steuerung nach einer ausreichenden Zeitreserve wieder zu übernehmen hat. Langfristig ist es dabei das Ziel mit Grad 5 die höchste Stufe der Automatisierung zu erreichen, bei der das Fahrzeug vollständig autonom betrieben werden kann und der Fahrer zu keiner Zeit mehr überwachend oder eingreifend tätig werden muss.

Zur Erzielung dieser höheren Automatisierungsgrade bedarf es Regelungssysteme, deren Toleranz gegenüber externen Störungen, unbekanntem Systemverhalten und Systemschäden über die von robusten Regelungsansätzen hinausgeht. Zu diesem Zweck werden Ansätze aus dem Feld der adaptiven Regelung verwendet, bei denen es dem Regelungssystem möglich ist, sich im laufenden Betrieb auf Grundlage der gemessenen Zustandsinformationen aktiv an die aktuelle Situation anzupassen. Auf diese Weise können sowohl Modell- und Parameterunsicherheiten als auch im gewissen Umfang schadensbedingte Systemdegradationen kompensiert werden, die bei Verwendung von linearen Regelungskonzepten zu einer Abnahme der Leistungsfähigkeit bis hin zum Verlust der Stabilität führen können. Beispiele solcher adaptiven Regelungsverfahren sind unter anderem Model Reference Adaptive Control (MRAC), L_1 Adaptive Control oder Adaptive Backstepping.

¹Das Bundesministerium für Verkehr und digitale Infrastruktur hat die entsprechende Verordnung am 18. Januar 2017 zur Kenntnis ins Bundeskabinett eingebracht. [1]

²Klassifizierung in Europa durch die Bundesanstalt für Straßenwesen und in den USA nach dem SAE International Standard J3016.

Eine über die aufgeführten Ansätze hinausgehende Möglichkeit zur Umsetzung eines adaptiven Regelungssystems mit theoretisch universeller Approximationsfähigkeit ist die Verwendung von Maschinenlernverfahren in Form künstlicher neuronaler Netzwerke (KNN) als Teilgebiet der künstlichen Intelligenz. Im Gegensatz zu den häufig im Kontext des automatischen Fahrens genannten Convolutional-Netzwerken (CNN) zur Umfeldwahrnehmung mit Deep-Learning-Methoden, sind KNNs deutlich weniger komplex aufgebaut und werden auf dieser Grundlage oftmals im aktiven Regelungsbetrieb weiter trainiert. Diese Online-trainierten Netzwerke werden dabei unter anderem zur Parameterschätzung oder der direkte Beeinflussen des Regelungssystems durch die Aktuatorik verwendet. Aus diesem Grund ist es von maßgeblicher Wichtigkeit sicherzustellen, dass die Netzwerkausgabe keinen destabilisierenden Einfluss auf das geschlossene Gesamtsystem hat und dass das dafür verantwortliche Netzwerktraining innerhalb stabiler Grenzen abläuft. Diese Forderung aufgreifend, soll die vorliegende Arbeit einen Beitrag zum sicheren Betrieb intelligenter und adaptiver Ansätze in sicherheitskritischen Regelungssystemen leisten und dies am Beispiel sowohl eines Flug- als auch eines Fahrzeugreglers demonstrieren.

1.1 Stand der Technik

Neuronale Netzwerke [2, 3] als ein Ansatz des Maschinenlernens werden in den letzten Jahren vor allem seit ihrer umfassenden Verwendung auf dem Gebiet der Computer-Vision maßgeblich mit der Forschung an künstlicher Intelligenz (KI) [4–6] in Verbindung gebracht. Dabei wurde die allgemeine Approximationsfähigkeit und die Fähigkeit zur Abbildung nichtlinearer Zusammenhängen von einfach aufgebauten Netzwerken bereits Ende der 1980er Jahre ausführlich dokumentiert [7–9]. Wissenschaftliche Anwendungsbereiche neuronaler Netzwerke sind dabei unter anderem Einsätze zur Parameterschätzung und Systemidentifikation [10–14] oder verschiedensten Aufgaben in der Mustererkennung wie der Analyse der Stromnetzauslastung [15], der Auswertung von Satellitenmessungen zur Polarlichtforschung [16] oder der Untersuchung von Mammographien [17]. Außerdem finden sich in der Literatur unterschiedliche Anwendungen zur neuronal gestützten und im laufenden Regelungsbetrieb umgesetzten Fehleridentifikation [18–20].

Die neuste Generation von neuronalen Netzwerken mit Fokus auf die umfängliche Nachbildung der biologischen Informationsverarbeitung bildet die Gruppe von gepulsten neuronalen Netzwerken (engl. Spiking-Neural-Networks - SNN). Dieser stark durch die Forschung in der Neurobiologie geprägte Ansatz nutzt neben der eigentlichen Ausgabe der Neuronen auch die Zeit der jeweiligen Ausgabeimpulse, um auf diese Weise Informationen zu übermitteln [21–23]. Der durch die möglichst exakte Umsetzung des biologischen Vorbilds resultierende Bedarf an Hochleistungsrechnern zur Erzielung sinnvoller Berechnungszeiten hat zu einer primär theoretischen Untersuchung dieser neuen Klasse von neuronalen Netzwerken geführt. Zur Bewältigung der benötigten Rechenanforderungen wird an sogenannter neuromorpher Hardware geforscht, die, anders als Von-Neumann-

Architekturen, nicht klassisch programmiert werden kann, sondern durch Lernverfahren in biologisch plausibler Art angepasst wird [24].

Der in den letzten Jahren in der populärwissenschaftlichen Literatur oftmals gleichgestellte Einsatz der Begrifflichkeiten KI und neuronale Netzwerke ist jedoch vor allem auf die rasant gestiegene Anzahl an Arbeiten zu Convolutional-Netzwerken zurückzuführen. Bei diesen aus dem Bereich des Deep-Learnings [25] stammenden Netzwerken werden große Mengen an Netzwerkeingaben in Abhängigkeit ihrer Beschaffenheit in sogenannten Feature-Maps zusammengefasst, was durch die mehrfache Wiederholung in hintereinander hängenden Schichten zu einer deutlichen Reduktion der zu verarbeitenden Datenmengen führt. Auf diese Weise wurde insbesondere in der Umfelderkennung und -wahrnehmung mit CNN-Ansätzen eine neue Qualität der Bildverarbeitung und des Situationsverständnisses realisiert [26–29]. So konnte beispielsweise im Jahr 2015 das Computerprogramm *AlphaGo* das erste Mal überhaupt mit Hilfe von Deep-Learning-Ansätzen einen Meister im asiatischen Brettspiel Go schlagen [30], was heute als Meilenstein der Forschung an künstlicher Intelligenz angesehen wird. Sich diesem Trend einer immer stärkeren Verbreitung von KI-Ansätzen in Forschung und Industrie anschließend, hat unter anderem die amerikanische Regierung Ende 2016 ein Dokument veröffentlicht, in dem direkte Empfehlungen an Behörden und Dritte ausgesprochen werden, die den flächendeckenden Einsatz von künstlicher Intelligenz vereinfachen und beschleunigen sollen [31].

Im Bereich der Flugregelung werden neuronale Netzwerke bereits seit Mitte der 1990er Jahre für die Anwendung in adaptiven Regelungskreisen untersucht und eingesetzt [32–37]. Dabei handelt es sich zumeist um die Kombination eines Reglers auf Basis dynamischer Inversion [38–40] mit Online-trainierten neuronalen Netzwerken, wie es auch in dieser Arbeit umgesetzt wird. Anders als bei den für die Umfelderkennung genutzten Convolutional-Netzwerken, liegt der Fokus dabei nicht auf der Verarbeitung tausender gleichzeitiger Eingangsinformationen, sondern auf dem schnellen und stabilen Ausgleich von sogenannten Inversionsfehlern. So konnte der bereits in verschiedenen robotischen Anwendungen [41–43] erprobte und auch schon für die Regelung von hochagilen Flugzeugen [44, 45] erfolgreich getestete Ansatz der nichtlinearen dynamischen Inversion mit der Erweiterung durch neuronale Netzwerke noch weiter in seiner Leistungsfähigkeit verbessert werden. Insbesondere auf dem Forschungsgebiet der unbemannten Flugsysteme sind zahlreiche Arbeiten zu diesem adaptiven Regelungsansatzes für sowohl Flächenflugzeuge als auch Hubschrauber und Multicopter entstanden [10, 46–51]. Dabei reicht die Spanne der Untersuchung von der Kompensation von Parameterunsicherheiten aufgrund günstiger Systemidentifizierungsmethoden im akademischen Bereich bis hin zu Technologiedemonstratoren für die spätere Übertragung auf manntragende Systeme. Letztere Anwendung ist insbesondere in Hinblick auf die Ausarbeitung von Zertifizierungs- und Zulassungskriterien für die Verwendung von neuronalen Netzwerken in sicherheitskritischen Systemen von besonderem Interesse [52, 53].

1.2 Gegenstand der Arbeit

Ziel der vorliegenden Arbeit ist es, durch die Entwicklung von auf der Gleitzustandsregelung (engl. Sliding-Mode-Control - SMC) erster und zweiter Ordnung basierender Trainingsverfahren und der damit realisierbaren Robustheits- und Transparenzsteigerung des Netzwerktrainings, einen Beitrag zur Verwendung von neuronalen Netzwerken in adaptiven Regelungssystemen zu leisten. Ausgangspunkt ist dabei eine detaillierte Beschreibung des Transfers der Gleitzustandsregelung auf das Training neuronaler Netzwerke und der Nachweis einer Überführung des Netzwerkfehlers in einen stabilen Gleitzustand. Anschaulich dargestellt wird dieses Verhalten und die erfolgreiche Anwendung der Gleitzustandsregelung als Lernverfahren für neuronale Netzwerke durch die Abbildung der Netzwerkzustände im Phasenraum.

Wo in früheren Arbeiten bereits demonstriert wurde, dass eine dynamische Lernratenbestimmung innerhalb berechneter Grenzen zu einem stabilen Gleitzustand erster Ordnung führen kann, wird in dieser Arbeit erstmals ein Konvergenzquotient eingeführt, mit dessen Hilfe, in Abhängigkeit des ebenfalls neu formulierten Intervallparameters, der Netzwerkzustand im kommenden Zeitschritt abgeschätzt werden kann. Dabei wird gezeigt, dass das eigentliche Schaltverhalten der als Trainingsverfahren umgesetzten Gleitzustandsregelung zum einen durch das Vorzeichen der Schaltfunktion und zum anderen durch die dynamische Lernratenbestimmung auf das neuronale Netzwerk übertragen wird. Sowohl für das Gleitzustandstraining erster als auch für das auf dem Twisting-Algorithmus basierende Verfahren zweiter Ordnung wird über umfassende Parameterstudien eine Aussage zur Wahl der neu eingeführten Netzwerkgrößen getroffen. Dabei ist es das Ziel der Arbeit einen allgemein gültigen Wertebereich zu identifizieren, so dass, durch den Wegfall der empirischen Lernratenauslegung in Kombination mit einer klaren Empfehlung für die neu definierten Parameter, die Auslegung eines Online-trainierten neuronalen Netzwerks deutlich vereinfacht und beschleunigt werden kann.

Ein weiterer Schwerpunkt liegt auf der Erweiterung eines nichtlinearen Inversionsregelkreises mit den gleitzustandstrainierten Netzwerken und den notwendigen Anpassungen der Beschränktheitsuntersuchung aufgrund möglicher negativer Lernraten als Ergebnis der angepassten Gewichtsänderungsvorschrift. Zur Darstellung der generischen Natur des vorgestellten Regelungskonzeptes werden zur Bewertung der Leistungsfähigkeit sowohl ein Flug- als auch ein Fahrzeugregler umgesetzt und jeweils in nichtlinearen numerischen Simulationen unter Einfluss unterschiedlicher Störgrößen untersucht. Besonderer Wert wird auf die umgesetzte Leistungsbewertung des Gleitzustandstrainings erster Ordnung im praktischen Fahrversuch gelegt. So kann die vorliegende Arbeit eine erstmalige Aussage zur Adaptionfähigkeit gleitzustandstrainierter Netzwerke in hochdynamischen Fahrsituationen treffen und dabei Vorteile gegenüber nicht-adaptiven Ansätzen aufzeigen.

1.3 Gliederung der Arbeit

Als Basis für die entwickelten Trainingsverfahren werden in Kapitel 2 die theoretischen Grundlagen künstlicher neuronaler Netzwerke mit Fokus auf dreischichtige und ausschließlich vorwärtsgerichteter Topologien dargestellt.

In Kapitel 3 wird im Anschluss das Prinzip und die Theorie der Gleitzustandsregelung erläutert und dabei auf die besonderen Eigenschaften eines Systems im Gleitzustand eingegangen. Als wesentliches Element für die in der vorliegenden Arbeiten entwickelten Lernverfahren wird darüber hinaus die Gleitordnung eingeführt und daran anknüpfend der Twisting-Algorithmus als Verfahren zweiter Ordnung beschrieben.

Als Kombination der bis zu diesem Punkt eingeführten Grundlagen werden in Kapitel 4 die entwickelten Gleitzustandstrainingsverfahren vorgestellt und hinsichtlich ihrer Konvergenzgeschwindigkeit und Stabilität bewertet. Dabei werden über die Verwendung von Erreichbarkeits- und Existenzbedingungen für die Ansätze erster und zweiter Ordnung jeweils Verfahren zur Berechnung dynamischer Lernratenintervalle vorgestellt.

Im Anschluss wird in Kapitel 5 das Regelungskonzept der nichtlinearen dynamischen Inversion mit Erweiterung um neuronale Netzwerke vorgestellt. Neben der Darstellung der Eingangs-/Ausgangslinearisierung steht dabei die Beschreibung der Fehlerdynamik des adaptiv erweiterten Systems und die für die Verwendung des Gleitzustandstrainings gegenüber bestehenden Ansätzen angepasste Beschränktheitsbetrachtung im Fokus.

In Kapitel 6 wird auf die Umsetzung des Regelungskonzepts als Flug- beziehungsweise Fahrzeugregler eingegangen. Dabei wird jeweils die zur Bewertung herangezogene Simulationsumgebung beschrieben und auf die Umsetzung von Parameterunsicherheiten, Nichtlinearitäten und aufgeschalteten Störgrößen eingegangen.

Anschließend werden in Kapitel 7 sowohl simulative Bewertungen der entwickelten adaptiven Regler unter Störeinflüssen und Einwirkung von Schadensszenarien dargestellt als auch eine Bewertung des vorgestellten Fahrzeugreglers in praktischen Fahrversuchen vorgenommen. Dabei wird unter anderem die Konvergenzgeschwindigkeit der Gleitzustandsverfahren untersucht und mit der des konventionellen backpropagation-Algorithmus verglichen.

Im darauffolgenden Kapitel 8 werden die zentralen Aspekte der vorliegenden Arbeiten noch einmal zusammengefasst. Dabei wird abschließend das Potential des adaptiven Regelungskonzepts mit Fokus auf die Leistungsfähigkeit der entwickelten Trainingsverfahren bewertet. Abgeschlossen wird die Arbeit mit einem kurzen Ausblick auf künftige wissenschaftliche Fragestellungen in Hinblick auf die Anwendung von Online-trainierten neuronalen Netzwerken.

Kapitel 2

Grundlagen neuronaler Netzwerke

Künstliche neuronale Netzwerke (KNN) verfolgen den Ansatz, spezielle Eigenschaften der Informationsverarbeitung, allem voran die Lernfähigkeit, eines biologischen neuronalen Netzwerks technisch zu modellieren. Dabei lassen sich nach [21] verschiedene Generationen von KNN definieren, die jeweils durch die Funktionsweise ihrer kleinsten Recheneinheit, den künstlichen Neuronen, definiert sind.

Bei den in dieser Arbeit verwendeten Netzwerken handelt es sich aufgrund der Nutzung von sogenannten Aktivierungsfunktionen und gradientenbasierten Trainingsverfahren um KNN der zweiten Generation. Dabei ist es, anders als bei den gepulsten neuronalen Netzwerken der dritten Generation, nicht das Ziel, das exakte biologische Vorbild und dessen Leistungsfähigkeit unter Verwendung von Hochleistungscomputern zu modellieren, sondern sich auf das grundlegende Prinzip der Informationsspeicherung und -propagation zu beschränken [54]. Gemeint ist damit die gewichtete Informationsweitergabe über hochgradig miteinander verbundener Neuronen. Der diesem Verfahren zu Grunde liegende Gedanke ist dabei relativ einfach technisch zu imitieren. Es werden eingehende Informationen nach ihrer Relevanz für die gewünschte Ausgabe des neuronalen Netzwerks bewertet (was der Gewichtung entspricht), innerhalb vernetzter und in Schichten angeordneten Neuronen aufsummiert und anschließend in einer Aktivierungsfunktion ausgewertet. Die auf diese Weise entstandenen Neuronenausgaben werden erneut gewichtet und an die nächste Schicht von verbundenen Neuronen übermittelt. Durch Anpassung der Verbindungsgewichte, was auch als Lernen oder Training bezeichnet wird, kann das gesuchte Eingangs-/Ausgangsverhalten des Netzwerks so angepasst werden, dass bei einer geeigneten Auslegung der Netzwerktopologie ein beliebig komplexer nichtlinearer Prozess approximiert werden kann. Folglich werden neuronale Netzwerke als universelle Approximatoren oder „Abbildungsmaschinen“ bezeichnet [2,7,8].

Die ersten Veröffentlichungen zu lernfähigen Algorithmen und zur technischen Umsetzung von Neuronen eines menschlichen Gehirns wurden 1943 von Warren McCulloch und Walter Pitts durch die Einführung eines künstlichen Neuronenmodells vorgestellt [55]. Ein nächster wichtiger Beitrag für die Erforschung von neuronalen Netzwerken war die

Vorstellung des sogenannten Perzeptrons durch Rosenblatt im Jahre 1958 [56]. Ausgehend von der These, dass Intelligenz durch die Verbindung vieler einfacher Recheneinheiten entsteht und dass Lernen durch die Veränderung dieser Verbindungen erreicht wird, veröffentlichte er ein erstes Lerngesetz für das Neuronenmodell von McCulloch und Pitts. In den anschließenden zwei Jahrzehnten wurde aufgrund mangelnder Computerleistung sowie dem Fehlglauben, dass komplexere und mehrschichtige Perzeptrons keine essentielle Verbesserung darstellen, die Förderung für Forschungsprojekte der künstlichen Intelligenz und der neuronalen Netzwerke vieler Orts gekürzt oder gänzlich eingestellt [6]. Erst Mitte der 80er Jahre, als klar wurde, dass Expertensysteme als Kontrast zu neurowissenschaftlichen Ansätzen komplex in der Umsetzung und letztlich in der Leistungsfähigkeit weit hinter den Erwartungen zurück blieben, erlebten neuronale Netzwerke eine Renaissance. So wurde der bereits 15 Jahre zuvor durch Bryson und Ho vorgestellte Backpropagation-Algorithmus [57] wieder aufgegriffen und parallel von verschiedenen Wissenschaftlern auf unterschiedliche Problemstellungen angewendet. Der wahre Durchbruch erfolgte jedoch durch die Veröffentlichung von Rumelhart und McClelland im Jahr 1986 über das Backpropagation-Training eines mehrschichtigen Perzeptrons [58]. Die damaligen Arbeiten sind noch heute Grundlage für viele Anwendungen neuronaler Netzwerke, so dass der allgemeine Backpropagation-Algorithmus auch in dieser Arbeit stets als Vergleich für die entwickelten Lernverfahren herangezogen wird.

Für die Eingliederung von neuronalen Netzwerken in den übergeordneten Forschungsbereich der künstlichen Intelligenz (KI) wird im folgenden Abschnitt ein verkürzter Überblick über die Entstehungsgeschichte und Definition dieser Forschungsdisziplin gegeben. Detaillierte Abhandlungen finden sich in [4–6].

2.1 Neuronale Netze im Feld der künstlichen Intelligenz

Die erste aufgezeichnete Verwendung des Begriffs künstliche Intelligenz (engl. Artificial-Intelligence) geht zurück auf das Jahre 1956, genauer auf eine Einladung zur Gründung einer Forschergruppe an der Universität Princeton. Mit dem heutigen Wissen zu den damaligen Forschungen wäre nach [4] der Begriff „berechnende Rationalität“ jedoch präziser gewesen und hätte die Unschärfe bei der Definition von Intelligenz umgehen können. Sich dieser Diskussion anschließend gibt es bis zum heutigen Tage keine klare und einheitliche Definition von künstlicher Intelligenz. So wurden über die Jahre verschiedene Ansätze diskutiert und vorgeschlagen, unter welchem Blickwinkel sich dem Feld der KI zu nähern ist und welche Methoden daraus abgeleitet werden können. Zusammengefasst sind dabei verschiedene Abstufungen der Eigenschaften *Denken* und *Handeln* in Bezug auf *menschliches* oder zwingend *rationales* Verhalten entstanden. Dabei ist die Unterscheidung von menschlichem und rationalem Verhalten nicht so zu verstehen, dass menschliches Handeln zwingend irrational ist. Es soll vielmehr der Unterschiedlichkeit menschlicher Leistungsfähigkeit gegenüber der streng auf Algorithmen und Vorschriftenden basierenden

Prozessierung durch Computer Rechnung getragen werden. In Tabelle 2.1 sind nach [4] verschiedene Ansätze in vier Quadranten aufgeteilt. Die linke Hälfte umfasst Definitionen der Imitation menschlicher Leistungen, während die rechte Hälfte Definitionen für ein ideales Vorgehen aufweist.

	Menschlich	Rational
Denken	<i>„Maschinen mit Verstand“</i> <i>„Automatisierung des Denkens“</i>	<i>„Studie mentaler Fähigkeiten“</i> <i>„Formalismen des Logischen“</i>
Handeln	<i>„Menschliche Funktion“</i> <i>„Computer für Tätigkeiten in denen der Mensch bislang überlegen ist“</i>	<i>„Intelligente Agenten“</i> <i>„Intelligentes Verhalten künstlicher Maschinen“</i>

Tabelle 2.1: Definitionen von künstlicher Intelligenz nach [4]

Die in dieser Arbeit untersuchten neuronalen Netzwerke, deren Training und letztlich ihre Anwendung in adaptiven Regelkreisen finden sich in der Definitionen des rationalen Handelns wieder. Verallgemeinert wird ein Verhalten gesucht, das zu einem in der gegebenen Zeit besten Ergebnis führt. Dabei ist anzumerken, dass für die Definitionen erneut, wie für die Bezeichnung der eigentlichen Disziplin KI selbst, der unscharf definierte Begriff der Intelligenz verwendet wird. Obwohl ein Jeder intelligentes Verhalten erkennen kann, ist es fragwürdig, ob dies zu einer präzisen Definition führt, die für die Bewertung von intelligenten Programmen verwendet werden kann¹. In [5] wird Intelligenz als Sammlung von Eigenschaften wie Verstehen, sinnvolles Verhalten, effektives Interagieren mit der Umwelt und nicht zuletzt Lernen beschrieben. Insbesondere die Fähigkeit zu lernen stellt einen schnell zu verstehenden Unterschied zu gewöhnlichen Rechenvorschriften dar. Wo nicht-intelligente Algorithmen programmiert sind, in einem bestimmten Umfeld, unter vorgegebenen Randbedingungen gezielte Berechnungen anzustellen, wird von intelligenten Agenten und Algorithmen erwartet, dass sie sich an ihre Umwelt anpassen, wie es unter anderem bei neuronalen Netzwerken der Fall ist. Zusammengefasst fällt die Arbeit an KNN und intelligenten Agenten somit klar in das Feld der KI, deckt dabei jedoch nur einen von vielen Schwerpunkten dieses vielfältigen Forschungsgebiets ab. Aktuelle Ergebnisse einer vom Umfang weit über die Forschung an rationalem Verhalten hinausgehenden Fusion der Disziplinen Neurowissenschaften und Computerwissenschaften unter dem Überbegriff der künstlichen Intelligenz, sind beispielsweise im Human-Brain-Projekt zu verfolgen [59].

In den folgenden Abschnitten wird die technische Umsetzung des biologischen Vorbilds so-

¹Im Duden ist beispielsweise von der „Fähigkeit des Menschen, abstrakt und vernünftig zu denken und daraus zweckvolles Handeln abzuleiten“ die Rede. Nach Duden - Die deutsche Rechtschreibung, Bibliographisches Institut, 26. Auflage, 2014

wie das Training neuronaler Netzwerke detailliert beschrieben. Dabei wird ein besonderer Fokus auf das Backpropagation-Training mehrschichtiger vorwärtsgerichteter Netzwerke gelegt, da dies die Grundlage für die in Kapitel 4 dargestellten Gleitzustandstrainingsverfahren sind.

2.2 Technische Umsetzung eines biologischen Vorbilds

Die beachtliche Leistungsfähigkeit des menschlichen Gehirns wird durch die massive Parallelität vieler mit einander verknüpften Recheneinheiten ermöglicht. Mit einer Anzahl von schätzungsweise 10^{11} Neuronen werden bis zu 10^{14} (100 Billionen) Verbindungen zwischen den einzelnen Neuronen gebildet [4]. Aufgebaut ist jedes dieser Neuronen durch einen Zellkörper, dem sogenannten Soma, der den Zellkern enthält sowie zwei Arten von Zellfortsätzen oder auch Fasern, die für die Reizweiterleitung verantwortlich sind. Zum einen handelt es sich dabei um die Dendriten, die sich aus dem Zellkörper heraus verzweigen und primär der Aufnahme von Eingangssignale dienen. Zum anderen besitzt jedes Neuron eine einzelne lange Faser, als Axon bezeichnet, das für die Weiterleitung des Ausgangsreizes an andere Neuronen verantwortlich ist. Die eigentlichen Verbindungen zwischen Axom und Dendriten stellen die Synapsen dar. Über diese können Dendriten mit 10 bis 100.000 anderen Neuronen verbunden sein [4]. Die Informationsübertragung innerhalb dieses stark verzweigten Systems aus Dendriten, Axonen und Synapsen erfolgt über einen komplexen elektrochemischen Prozess, der unter anderem in [2] detailliert beschrieben ist.

Für die technische Umsetzung dieses biologischen Vorbilds existieren heutzutage komplexe Forschungsvorhaben wie zum Beispiel der Simulation des Cerebellums (Kleinhirns) [60] oder der digitalen Rekonstruktion und Simulation eines Teils des Neocortex [61]. Diese Ansätze werden jedoch nicht mehr zum Gebiet der künstlichen neuronalen Netzwerke gezählt und sollen aus diesem Grund hier nicht weiter vertieft werden. Ist es das Ziel, das Prinzip der gewichteten und lernfähigen Informationsverarbeitung auf eine Weise umzusetzen, die unter anderem ihre Anwendung in hochdynamischen Regelungssystemen erlaubt, müssen zwangsläufig Vereinfachungen getroffen werden. Die oben beschriebene Entwicklungsgeschichte von neuronalen Netzwerken hat schließlich zu einem Verfahren geführt, welches sich die zwei grundlegenden Merkmale des biologischen Vorbilds zunutze macht, technisch aber noch zu beherrschen ist. So erwirbt ein Netzwerk aus miteinander verbundenen Recheneinheiten durch einen Lernprozess Wissen aus seiner Umgebung und nutzt die Gewichtung der Neuronenverbindung dafür, das erlernte Wissen zu speichern [3]. Sind diese beiden Aspekte gegeben, kann ein neuronales Netz, als eindeutiger Kontrast zum menschlichen Gehirn, aus nur wenigen Neuronen bestehen und dennoch Wissen akkumulieren und dieses in den Verbindungen speichern.

2.2.1 Das künstliche Neuron

Die gängige Umsetzung von neuronalen Netzwerken besteht aus in Schichten angeordneten Neuronen. Dabei wird die jeweils betrachtete Schicht mit dem Index (l) und die Anzahl der in dieser Schicht vorhandenen Neuronen mit $q^{(l)}$ bezeichnet. Das Verbindungsgewicht $w_{ij}^{(l)}$ zwischen zwei Neuronen wird jeweils der Schicht des Startneurons zugeordnet. Hierbei steht i für den Index des Start- und j für den Index des Zielneurons. Die Eingabe eines Neurons, auch Aktivierungszustand $n_j^{(l)}$ genannt, ergibt sich aus der Summe der Neuronenausgänge der vorangegangenen Schicht $o_i^{(l-1)}$ multipliziert mit den jeweiligen Verbindungsgewichten. Für den Neuroneneingang folgt somit:

$$n_j^{(l)} = \sum_{i=1}^{q^{(l-1)}} o_i^{(l-1)} \cdot w_{ij}^{(l-1)} . \quad (2.1)$$

Über eine sogenannte Transfer- oder auch Aktivierungsfunktion $f_j^{(l)}$ wird innerhalb des Neurons die Neuroneneingabe $n_j^{(l)}$ in die Neuronenausgabe $o_j^{(l)}$ umgesetzt. Es folgt:

$$o_j^{(l)} = f_j^{(l)}(n_j^{(l)}) . \quad (2.2)$$

In Abbildung 2.1 ist ein beispielhaftes künstliches Neuron mit Verbindungen zu drei Neuronen der vorangegangenen $(l-1)$ und drei Verbindungen zur folgenden Schicht $(l+1)$ dargestellt.

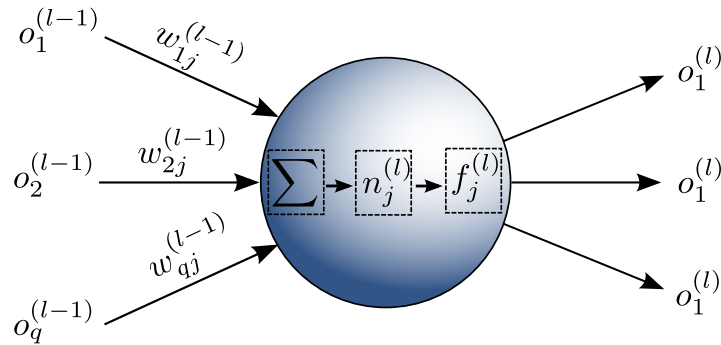


Abbildung 2.1: Darstellung eines vernetzten künstlichen Neurons mit drei Ein- und Ausgabegrößen.

Für die Abbildung nichtlinearer Zusammenhänge durch das neuronale Netz ist es zwingend notwendig, dass zumindest eine nichtlineare Transferfunktion innerhalb des Netzwerks existiert. Aus biologischer Sicht handelt es sich bei diesen Funktionen um die Berechnung des Aktivitätslevels des entsprechenden Neurons. In der ersten Generation von neuronalen Netzen, noch mit einer nicht stetigen Schwellenwertfunktion dargestellt, definiert die Transferfunktion, ob das Neuron als Folge der eintreffenden Neuroneneingänge selber "feuert" oder gehemmt bleibt. In der Literatur finden sich verschiedenste Arten von Transferfunktionen, wobei typischerweise für Neuronen einer Schicht einheitliche Funktionen gewählt werden [2,62]. In Abbildung 2.2 sind die ideale Schwellenwertfunktion (a), die für neuronale Netze

der zweiten Generation typische sigmoide (S-förmige) Tangens Hyperbolicus-Funktion (b) sowie die lineare Identitätsfunktion (c) dargestellt.

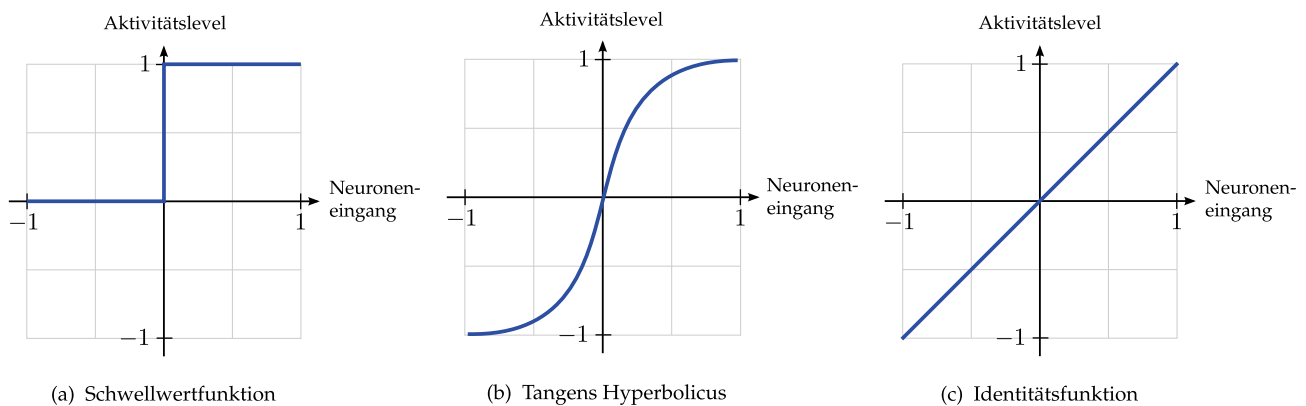


Abbildung 2.2: Schematische Darstellung verschiedener Transferfunktionen

Sowohl der Tangens Hyperbolicus als auch die Schwellwertfunktion können dabei als biologisch plausibel bezeichnet werden, da ihre Ausgänge in beide Richtungen begrenzt sind. Dies entspricht ungefähr dem Verhalten im menschlichen Gehirn, bei dem auch nur ein begrenzter Sprung im Aktionspotential von ca. -40 bis -80mV auf maximal +55mV entsteht. Die höchste Plausibilität weist dabei die Schwellwertfunktion auf, da das Aktionspotential von biologischen Neuronen immer in gleicher Stärke vorliegt, unabhängig von der Höhe der Schwellwertüberschreitung [62]. Für das in Abschnitt 2.3 dargestellte Lernverfahren ist es jedoch zwingend notwendig eine stetig differenzierbare Transferfunktion zu wählen, so dass die biologische Plausibilität letztendlich nur von geringerer Bedeutung ist und statt der unstetigen Schwellwertfunktion der stetige Tangens Hyperbolicus verwendet wird. Die Identitätsfunktion eignet sich vor allem für die Ein- und Ausgabeschichten eines Netzwerks, da sie nicht auf einen beschränkten Wertebereich abbildet und somit einen größeren Zustandsraum an sowohl Ein- als auch Ausgängen nutzbar macht.

2.2.2 Netzwerktopologien

Mit dem oben beschriebenen Neuronenmodell und den enthaltenen Transferfunktionen ist es nun möglich, ein vollständiges neuronales Netzwerk umzusetzen. Wie bereits beschrieben, ist es dabei üblich die Neuronen in Schichten anzuordnen. Die genaue Art und Weise der Anordnung in Bezug auf die Anzahl der Schichten, der Neuronen und die Art der Verbindung wird als Topologie bezeichnet. Abhängig vom Einsatzzweck, dem Trainingsziel und der zur Verfügung stehenden Rechenkapazität existiert eine Vielfalt an verschiedenen Netzwerktopologien. An dieser Stelle wird ausschließlich auf eine Auswahl davon eingegangen. Für tiefer gehende Informationen sei unter anderem auf [2, 3, 62] verwiesen.

Die im Anschluss dargestellten Topologien unterscheiden sich einzig in ihrer Art der Neuronenverbindung. Die Architektur, also die Anzahl der Schichten und den darin enthaltenen Neuronen, ist für alle Beispiele identisch. Es handelt sich dabei um dreischichtige Netze mit drei Neuronen in der Eingabeschicht, vier Neuronen in der verdeckten Schicht und einem Neuron in der Ausgabeschicht. Neben einer grafischen Darstellung der verbundenen Neuronen wird für jedes Beispiel die Hinton-Darstellung oder auch Verbindungsmatrix gezeigt [62–64]. Dabei handelt es sich um ein Hilfsmittel, um auf abstrakte Weise einen schnellen Überblick über die Netzwerktopologie zu erhalten. Die erste Spalte der Verbindungsmatrix enthält die Auflistung aller Startneuronen und die erste Zeile der Matrix die Auflistung aller Zielneuronen. Durch die farbige Markierung eines Feldes der Matrix kann jeweils direkt das Start- und das Zielneuron einer Verbindung identifiziert werden. So ist es möglich, durch die Darstellung der Verbindungsgeometrie, Netzwerke direkt miteinander zu vergleichen, ohne die vollständig Abbildung von verbundenen Neuronen zu benötigen.

Feedforward-Netzwerke

Ausschließlich vorwärtsgerichtete neuronale Netzwerke werden als Feedforward-Netzwerke bezeichnet. Dabei kann unterschieden werden, ob nur ausgewählte oder sämtliche Neuronen einer Schicht mit den Neuronen der Folgeschicht verbunden sind. Durch den einfachen Aufbau und dem somit vergleichsweise geringem Bedarf an Rechenkapazität ist diese Topologie ideal für das Training im laufenden Regelungsbetrieb (Online-Training, vergleiche Abschnitt 2.3). In Abbildung 2.3 ist sowohl die Darstellung als auch die Verbindungsmatrix eines dreischichtigen Feedforward-Netzwerks abgebildet. Charakteristisch für die Darstellung von feedforward Netzwerken in einer Verbindungsmatrix sind die ausschließlich oberhalb der Hauptdiagonalen verlaufenden Markierungen.

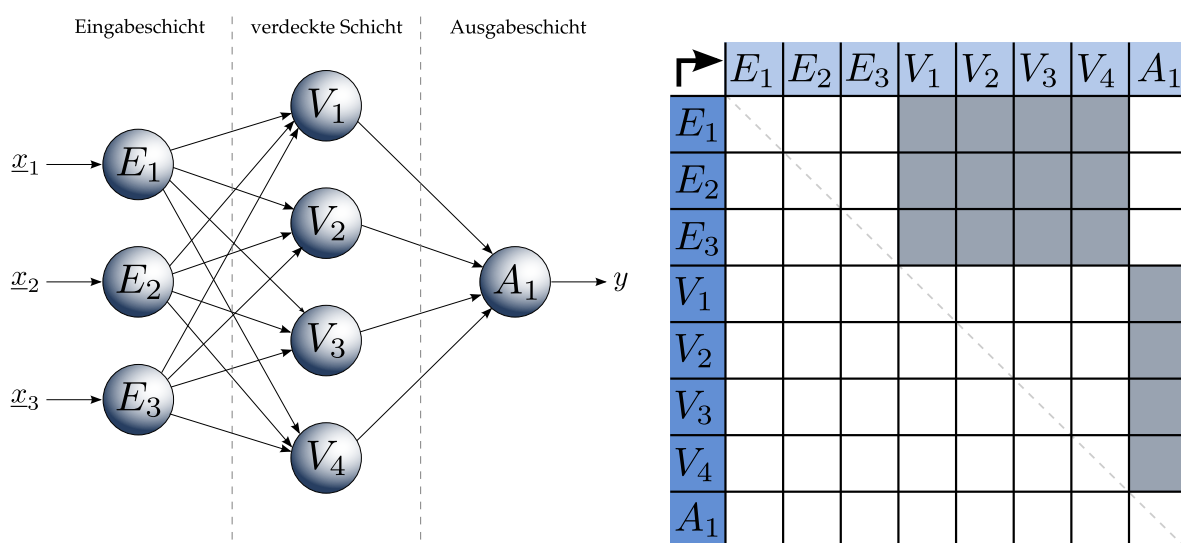


Abbildung 2.3: Feedforward-Netzwerk mit Verbindungsmatrix

Shortcut-Netzwerke

Abgeleitet vom englischen shortcut, zu Deutsch Abkürzung, enthalten Shortcut-Netzwerke Verbindungen, die eine Schicht überspringen. In dem hier dargestellten Beispiel sind die Neuronen der Eingabeschicht somit direkt mit dem einen Neuron in der Ausgabeschicht verbunden. Dabei handelt es sich um eine Sonderform des Feedforward-Netzwerks, da noch immer ausschließlich vorwärtsgerichtete Verbindungen existieren. Durch die zusätzlichen Verbindungen erhält das Netzwerk weitere Freiheitsgrade im Training, zeitgleich wird das Training jedoch auch aufwändiger, was an späterer Stelle in dieser Arbeit, bei der Beschreibung des Backpropagation-Algorithmus noch deutlich wird. In Abbildung 2.4 sind sowohl die Struktur als auch die Verbindungsmatrix eines beispielhaften shortcut Netzwerks zu sehen. Dabei sind die im Vergleich zum reinen Feedforward-Netzwerk aus Abbildung 2.3 neu dazugekommenen Verbindungen schwarz und die bisherigen in grau dargestellt. In der Verbindungsmatrix sind die drei neuen Shortcut-Verbindungen ebenfalls hell und die bisherigen Verbindungen in dunkel dargestellt.

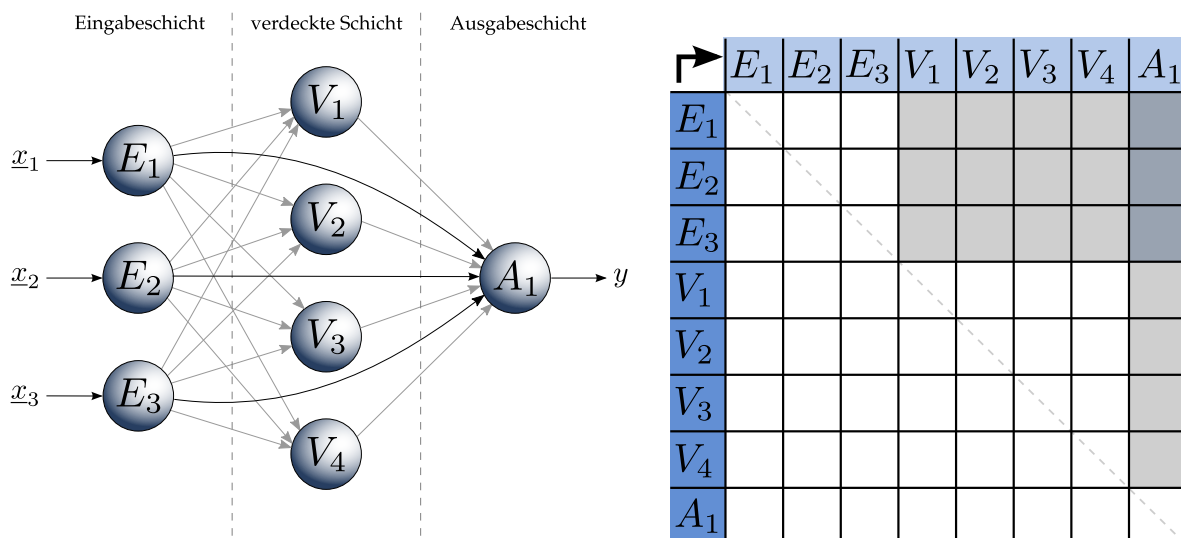


Abbildung 2.4: Shortcut-Netzwerk mit Verbindungsmatrix

Direct-Recurrent-Netzwerke

Recurrent-Netzwerke besitzen anders als reine Feedforward-Netzwerke rückgerichtete Verbindungen. Dabei wird zwischen direkten Rückkopplungen eines Neurons mit sich selber, indirekten Rückkopplungen mit Neuronen vorangegangener Schichten und lateraler Rückkopplung innerhalb einer Schicht unterschieden. Durch die Rückführung einer durch ein Neuron bereits verarbeiteten Information ist es dem Recurrent-Netzwerk möglich, eine Art zeitliches Gedächtnis zu entwickeln. So können sequenzielle Eingangsinformationen oder allgemein Zeitinformationen dynamischer Systeme erfasst und verarbeitet werden. In einem Direct-Recurrent-Netzwerk ist jedes Neuron über eine Schleife mit sich selbst verbunden, so dass es sich in Abhängigkeit der Ausgabe selbst hemmt oder verstärkt. In

Abbildung 2.5 ist ein Direct-Recurrent-Netzwerk und die dazugehörige Verbindungsmatrix dargestellt.

Indirect-Recurrent-Netzwerke

Wo bei Direct-Recurrent-Netzwerken nur eine Verbindung und somit ein zusätzliches Gewicht pro Neuron ergänzt wird, besitzt ein Indirect-Recurrent-Netzwerk neben den Feedforward-Verbindungen eine vollständige Feedbackward-Verknüpfung. Die Anzahl der Neuronen verdoppelt sich somit gegenüber der nur vorwärtsgerichteten Topologie. Charakteristisch in der Verbindungsmatrix ist dafür die Spiegelung entlang der Hauptdiagonale, wie sie in Abbildung 2.6 dargestellt ist.

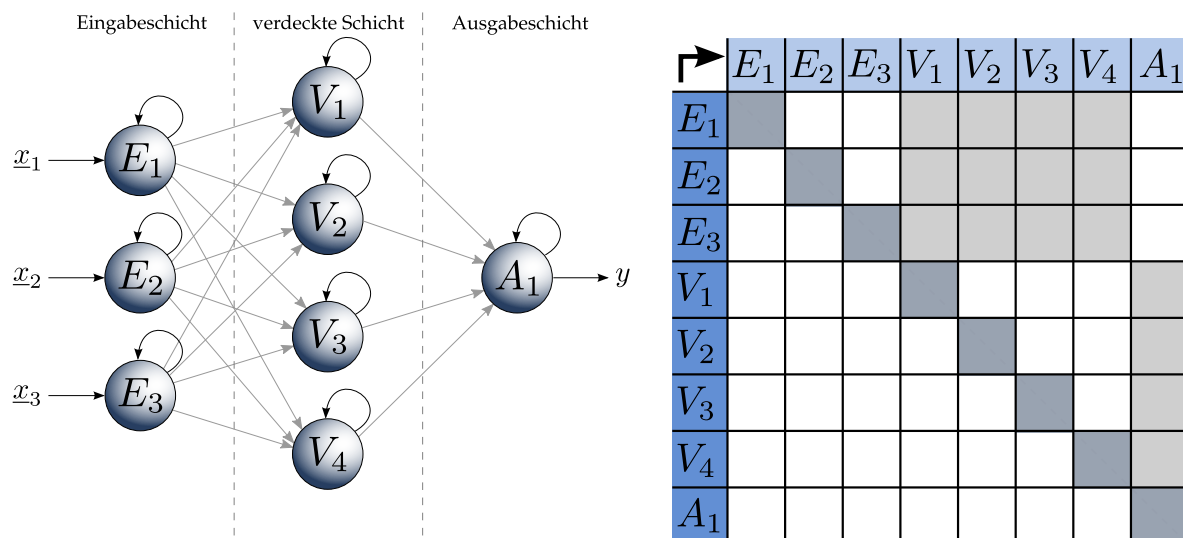


Abbildung 2.5: Direct-Recurrent-Netzwerk mit Verbindungsmatrix

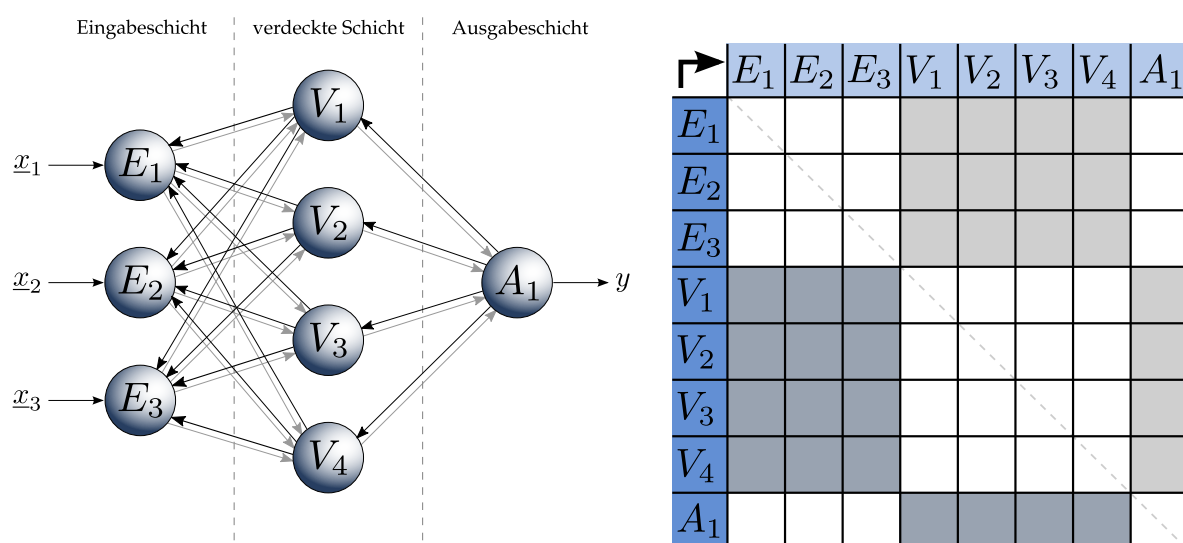


Abbildung 2.6: Indirect-Recurrent-Netzwerk mit Verbindungsmatrix

Für die in dieser Arbeit dargestellten Herleitungen und Untersuchungen wird sich, aufgrund ihrer besonderen Eignung zum Online-Training, ausschließlich auf Feedforward-Netzwerke beschränkt. Darüber hinaus ist die Anwendung von vorwärts gerichteten neuronalen Netzen zur Erweiterung von Inversionsreglern umfassend untersucht worden, so dass bei der Umsetzung der im Anschluss vorgestellten neuartigen Lernverfahren auf bestehende Theorien aufgebaut werden kann.

Eine in dieser Arbeit genutzte Erweiterung der in Abbildung 2.3 dargestellten Feedforward-Topologie besteht aus den sogenannten Biasneuronen. Bei diesen besonderen Elementen handelt es sich um Neuronen mit einer konstanten Ausgabe von „1“, die entsprechend der Feedforward-Topologie mit allen Neuronen der nachfolgenden Schicht verbunden sind. Hintergrund der Nutzung von Biasneuronen ist die Umsetzung eines Schwellenwertes außerhalb der Transferfunktion. So wird durch die gewichtete Verbindung eines konstanten Wertes zu den nachfolgenden Neuronen die Einrechnung des Schwellenwertes von der Aktivierungsfunktion in die Propagationsberechnung verschoben. Mit anderen Worten, der Schwellenwert wird einfach von den übrigen eintreffenden Neuroneneingaben subtrahiert, anstatt innerhalb der Transferfunktion berücksichtigt zu werden [63]. In Abbildung 2.7 ist ein dreischichtiges Feedforward-Netzwerk mit zwei Biasneuronen, der Identitätsfunktion in Ein- und Ausgabeschicht, sowie der Tangens Hyperbolicus-Funktionen in der verdeckten Schicht dargestellt.

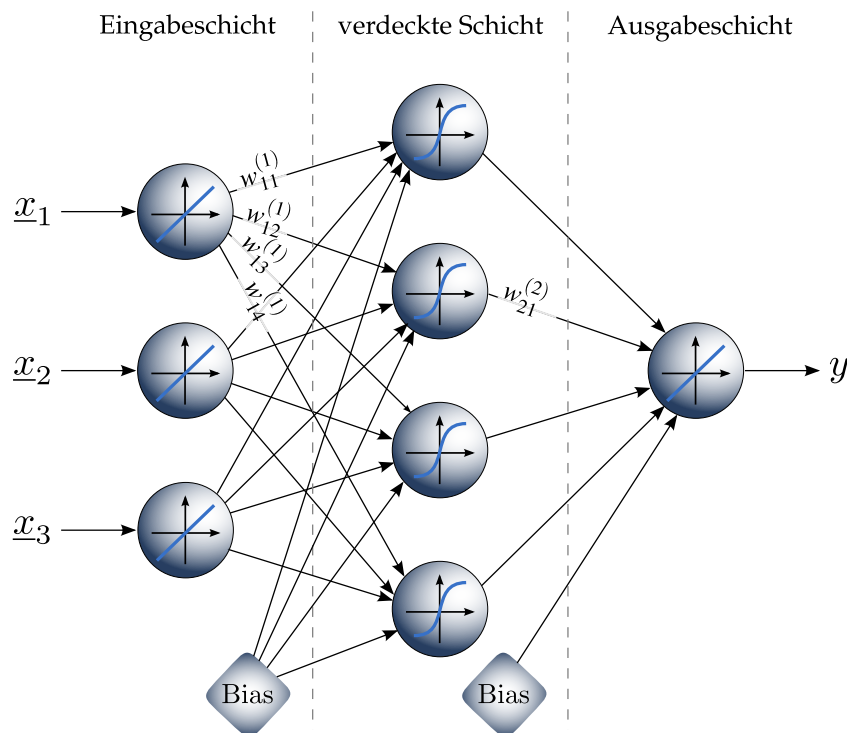


Abbildung 2.7: Darstellung eines vorwärtsgerichteten Netzes mit linearen Identitätsfunktionen in der Ein- und Ausgabeschicht, sowie sigmoider Funktion in der einen verdeckten Schicht.

Hier sind die Eingangsgrößen x_q Einträge des Eingangsvektors \vec{x} und die Ausgabe der Skalar y . Mit Ausnahme der abweichenden Anzahl von Neuronen in der Eingabe- und der verdeckten Netzwerkschicht, handelt es sich dabei um die Topologie, wie sie bei den in dieser Arbeit angestellten Untersuchungen verwendet wird. Für ein auf diese Weise aufgebautes Netzwerk mit jeweils q Neuronen in den drei Schichten lassen sich in Einklang mit der oben vorgestellten Indizierung zwei allgemein gültige Gewichtsmatrizen aufstellen. Dabei werden die Biasneuronen mit b indiziert. Es folgt:

$$\mathbf{w}^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \dots & w_{1q^{(2)}}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & \dots & w_{2q^{(2)}}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{q^{(1)}1}^{(1)} & w_{q^{(1)}2}^{(1)} & \dots & w_{q^{(1)}q^{(2)}}^{(1)} \\ w_{b1}^{(1)} & w_{b2}^{(1)} & \dots & w_{bq^{(2)}}^{(1)} \end{bmatrix}, \quad \mathbf{w}^{(2)} = \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} & \dots & w_{1q^{(3)}}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} & \dots & w_{2q^{(3)}}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{q^{(2)}1}^{(2)} & w_{q^{(2)}2}^{(2)} & \dots & w_{q^{(2)}q^{(3)}}^{(2)} \\ w_{b1}^{(2)} & w_{b2}^{(2)} & \dots & w_{bq^{(3)}}^{(2)} \end{bmatrix}. \quad (2.3)$$

Im vorliegenden Fall mit nur einem Ausgabeneuron in der Ausgabeschicht des Netzwerks vereinfacht sich der Ausdruck für die Gewichtsmatrix $\mathbf{w}^{(2)}$ zu einem Spaltenvektor. Für eine vereinfachte und übersichtlichere Schreibweise im weiteren Verlauf lassen sich die beiden einzelnen Gewichtsmatrizen zu einer allgemeinen Gewichtsmatrix \mathbf{w} zusammenfassen.

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}^{(2)} & 0 \\ 0 & \mathbf{w}^{(1)} \end{bmatrix} \quad (2.4)$$

Mit Hilfe der Gewichtsmatrizen und den Gleichungen 2.1 und 2.2 lässt sich die Netzausgabe eines Feedforward-Netzwerks in allgemeiner Vektorschreibweise angeben. Für das in Abbildung 2.7 dargestellte dreischichtige Netzwerk mit linearen Identitätsfunktionen in Ein- und Ausgabeschicht kann dabei ein vereinfachter Ausdruck gewählt werden. So kann die Neuroneneingabe der ersten Schicht $\vec{n}^{(1)}$ direkt als Eingabevektor \vec{x} und die Ausgabe $o_1^{(3)}$ der Ausgabeschicht als Netzausgang y notiert werden. So vereinfacht folgt:

$$y = \mathbf{w}^{(2)} \cdot \vec{f}^{(2)} \left(\mathbf{w}^{(1)} \cdot \vec{x} \right). \quad (2.5)$$

Da es sich bei der Findung einer geeigneten Topologie, insbesondere für die Abbildung von dynamischen Prozessen, meistens um einen iterativen und zeitintensiven Prozess handelt, wird in [48] ein systematischer und ingenieurtechnisch reproduzierbarer Entwurfsprozess neuronaler Netzwerke vorgeschlagen. Durch die statistische Auswertung und Bewertung von großen Gruppen von trainierten Netzwerken können Aussagen über die geeignete Anzahl von Schichten und Neuronen, die zu verwendenden Transferfunktionen und Eingangsgrößen getroffen werden. Darüber hinaus soll durch die große Menge an trainierten Netzwerken der häufig geäußerten Kritik der nicht nachvollziehbaren Arbeitsweise von neuronalen Netzwerken von statistischer Seite widersprochen werden.

2.3 Lernverfahren

Wie bereits beschrieben, wird das Lernen oder auch Training eines neuronalen Netzes gewöhnlich durch das gezielte Anpassen der Verbindungsgewichte zwischen den einzelnen Neuronen durchgeführt. Dabei wird das Wissen des Netzwerks durch die Gewichtung der einzelnen Netz- und Neuroneneingänge dargestellt. Darüber hinaus existieren jedoch noch weitere Möglichkeiten, die Anpassung eines Netzwerks an die Umgebung und somit den Lernprozess abzubilden. Neben der Gewichtsänderung werden in [62] beispielhaft noch folgende weitere Ansätze aufgeführt:

- Aufbau neuer Neuronenverbindung
- Entfernung bestehender Neuronenverbindung
- Veränderung der Transferfunktionen
- Schichterweiterung um neue Neuronen
- Entfernung von bestehenden Neuronen

Innerhalb dieser Arbeit wird sich jedoch auf das Anpassen der Verbindungsgewichte zur Erzeugung einer gewünschten Netzausgabe beschränkt. Der Lernprozess kann somit, wie in Gleichung 2.6 zu sehen, als Aktualisierung der aktuellen Gewichtsmatrix um ein Gewichtsincrement $\Delta \mathbf{w}$ ausgedrückt werden.

$$\mathbf{w}_{neu} = \mathbf{w}_{alt} + \Delta \mathbf{w} \quad (2.6)$$

Das prinzipielle Vorgehen bei der Berechnung der Gewichtsänderung und somit die Art des Trainings kann in verschiedene Ansätze klassifiziert werden. Zu unterscheiden sind dabei unüberwachte und überwachte Verfahren, sowie bei letzteren noch die Unterscheidung in korrigierendes oder rein verstärkendes Lernen [2, 3, 62]. Beim unüberwachten Lernen erhält der Trainingsalgorithmus keinerlei Information über die Wunschausgabe des Netzwerks. Stattdessen werden Ähnlichkeiten in den Eingabegrößen \vec{x} beziehungsweise $n_j^{(l)}$ dazu genutzt Rückschlüsse auf die Wichtigkeit der einzelnen Verbindungen zu ziehen. Ein bekanntes unüberwachtes Lernverfahren ist die bereits 1949 von Donald O. Hebb vorgestellte Hebb-Regel [65]. Sie besagt, dass das Gewicht zweier verbundener Neuronen in dem Falle verändert wird, dass die Ausgabe des ersten Neurons zu einer Ausgabe des zweiten Neurons führt, so dass beide Neuronen gleichzeitig aktiv sind. Nach Hebb entspricht dies dem Wachstumsprozess oder auch der metabolischen Veränderung, welche die Effizienz zweier so miteinander interagierender Neuronen erhöht.

Im Vergleich dazu wird bei den beaufsichtigten Lernverfahren eine Aussage darüber getroffen, inwieweit die aktuelle Netzausgabe einen Trainingserfolg oder -misserfolg darstellt. Bei einem rein verstärkenden Trainingsverfahren wird dabei keine quantitative Aussage getroffen, wieweit die Ausgabegröße von der Zielgröße entfernt liegt. Vielmehr wird die

Ausgabe in gut oder schlecht klassifiziert und das Training danach angepasst. Bei den überwachten und korrigierenden Lernverfahren, wie sie auch in dieser Arbeit Anwendung finden, wird der Fehler zwischen Netzausgabe und Zielgröße exakt bestimmt und für die Anpassung der Netzwerkgewichte angewendet. So kann neben der Information über die reine Tendenz auch direkt eine Aussage über die Qualität des aktuellen Netzwerktrainings getroffen werden.

Eine für die Anwendung der neuronalen Netzwerke wichtige Unterscheidung ist die Anzahl der für eine Gewichtsänderung verarbeiteten Datensätze oder auch Trainingsmuster. Werden die Netzwerkgewichte ausschließlich in Abhängigkeit des aktuellen Ausgabefehlers angepasst, so dass über die Zeit ein sequentielles Training des Netzwerks erfolgt, wird vom sogenannten Online-Learning gesprochen [2]. Durch die Eigenschaft, in Echtzeit die Verbindungsgewichte anpassen und somit auf spontane Änderungen von Eingangs- und Trainingsgrößen reagieren zu können, sind Online-trainierte Netzwerke vor allem für die regelungstechnische Anwendung geeignet. Durch die Einbindung in einen geschlossenen Regelkreis ist es dabei jedoch von besonderer Wichtigkeit, dass das Netzwerktraining stabil abläuft und das Regelungssystem nicht durch eine destabilisierende Netzwergausgabe an Robustheit verliert.

Im Gegensatz dazu werden beim Offline-Learning oder genauer noch beim Batch-Learning die Netzwerkgewichte nicht in Echtzeit sondern über die Vorgabe von im Vorfeld angelegten Trainingsmustern angepasst. Dabei ist es üblich, die Netzwerkgewichte nicht nach jedem Trainingsmuster anzupassen, sondern mehrere Muster zusammengefasst zu verarbeiten. Das Training über alle verfügbaren Muster wird Trainingsepoche genannt, so dass mit Variation der Reihenfolge, in welchem dem Netzwerk die Muster präsentiert werden, ein Netzwerk häufig über mehrere Epochen Offline-trainiert wird [2, 48]. Eine Möglichkeit zur Kombination von Online und Offline-Learning ist das Vortraining von Netzwerkgewichten auf Basis aufgezeichneter Daten, um diese Gewichtungskonfiguration im Online-Betrieb weiter anzupassen. Auf diese Weise können Online-Learning-Netzwerke mit einem gewissen Vorwissen initialisiert werden. Innerhalb dieser Arbeiten werden aufgrund der ausschließlich regelungstechnischen Verwendung der Netzwerke nur Verfahren des Online-Learnings betrachtet. Das bekannteste dieser Verfahren ist der bereits mehrfach angesprochene Backpropagation-Algorithmus [2, 57, 58]. In den folgenden Abschnitten sollen die dem Verfahren zu Grunde liegende Theorie, die Mathematik sowie die Ansatzpunkte für mögliche Erweiterungen und Verbesserungen des Verfahrens durch die in dieser Arbeit vorgeschlagenen Methoden dargestellt werden. Die Ausarbeitung orientiert sich dabei an den vorangegangenen Untersuchungen aus [48, 66].

2.3.1 Backpropagation-Algorithmus

Der essentielle Aspekt des in Gleichung 2.6 dargestellten Lernprozesses ist die Berechnung der Gewichtsänderung Δw , da sie das Ergebnis des Trainingsalgorithmus darstellt und so-

mit das Ziel verfolgt, die Netzwerkausgabe dem vorgegebenen Zielwert anzugleichen. Um dies zu gewährleisten, benötigt der Backpropagation-Algorithmus eine Aussage über die aktuelle Qualität der Netzausgabe in Hinblick auf das Trainingsziel. Zu diesem Zweck wird oftmals der quadratische Fehler E zwischen der kommandierten Ausgabe \vec{y}_k und der aktuellen Ausgabe \vec{y} verwendet.

$$E = \frac{1}{2} (\vec{y}_k - \vec{y})^T \cdot (\vec{y}_k - \vec{y}) . \quad (2.7)$$

Um das Netzwerk nun zielgerichtet trainieren zu können, bedarf es noch der Information, in wieweit sich die Fehlergröße E durch die gezielte Gewichtsveränderung beeinflussen lässt. Hierfür greift der Backpropagation-Algorithmus auf ein klassisches Werkzeug der Optimierung, den Gradientenabstieg, zurück. Über die Bildung der partiellen Ableitung der Fehlerfunktion nach allen Netzwerkgewichten wird ein Gradient berechnet, entlang dem das lokale Minimum der Fehlerfunktion gefunden werden kann. Da für die Berechnung des Gradienten $q + 1$ gewichtete Verbindungen zwischen allen l Schichten berücksichtigt werden müssen, handelt es sich dabei um ein hoch dimensionales Optimierungsproblem. Zur Verdeutlichung ist in Abbildung 2.8 eine beispielhafte Fehlerfunktion E über zwei Gewichte aufgetragen. Es ist deutlich zu erkennen, dass bei der Gewichtungskonfiguration von $w_1 = 2$ und $w_2 = -8$ das Minimum der hier dargestellten Fehlerfunktion erreicht und somit das Training optimal ausgeführt wäre.

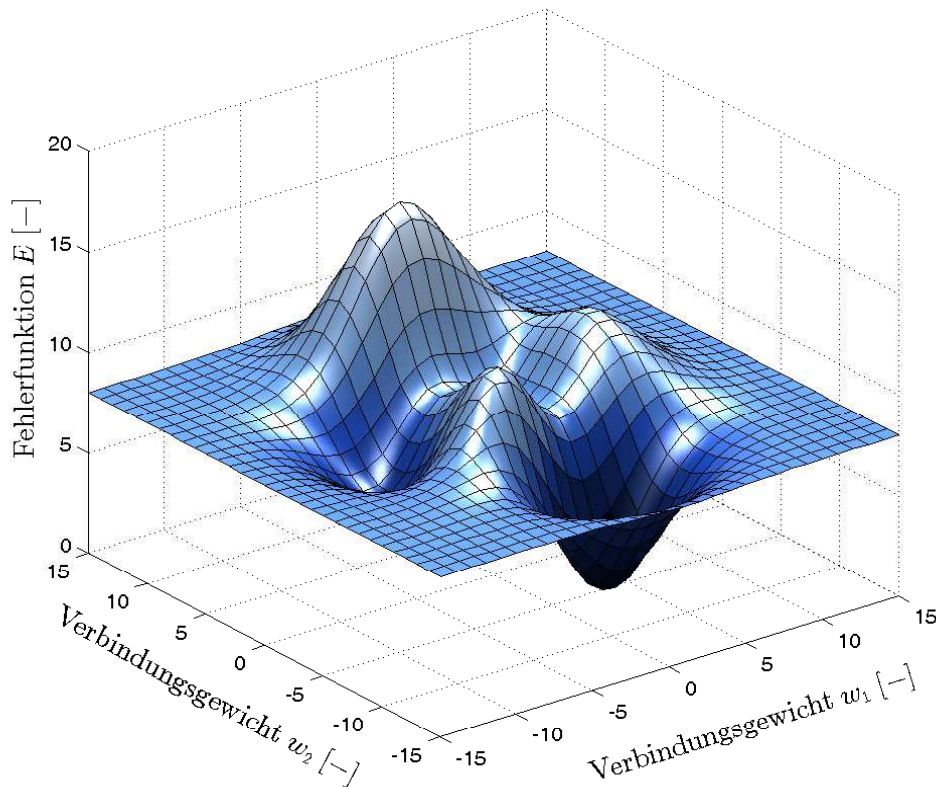


Abbildung 2.8: Beispielhafte Fehlerfunktion für die Berechnung eines Gradienten zur Findung des Minimums durch den Backpropagation-Algorithmus.

Dem üblichen Verfahren eines Gradientenabstiegs entsprechend, muss auch für den Backpropagation-Algorithmus eine Schrittweite gewählt werden, mit der entlang dem negative Gradienten das Minimum gefunden werden soll. Auf das Training eines neuronalen Netzwerks bezogen, wird diese Größe als Lernrate μ bezeichnet. Durch ihre Eigenschaft, das Training zu beschleunigen oder abzubremesen, stellt sie für den Trainingserfolg einen elementar wichtigen Parameter dar. Am Ende dieses Kapitels wird detailliert auf die Problematik einer empirischen Lernratenwahl eingegangen, was unter anderem als Motivation für die in dieser Arbeit vorgestellten, neuartigen Lernverfahren dient.

Mit dem quadratischen Fehler E , dem Nabla-Operator ∇ für die Zusammenfassung aller partiellen Ableitungen und der Lernrate μ kann das Ergebnis des Backpropagation-Algorithmus $\Delta \mathbf{w}$ wie folgt geschrieben werden:

$$\Delta \mathbf{w} = -\mu \cdot \frac{\partial E}{\partial \mathbf{w}} = -\mu \cdot \nabla E(\mathbf{w}) . \quad (2.8)$$

Ein Nachteil dieser oftmals umgesetzten Variante des Backpropagation-Algorithmus mit Nutzung des quadratischen Fehlers ist der Wegfall der Vorzeicheninformation des eigentlichen Netzwerkfehlers (auch Ausgabefehler genannt) $\vec{\epsilon}$ als Differenz zwischen dem kommandierten \vec{y}_k und dem aktuellen Ausgang \vec{y} .

$$\vec{\epsilon} = \vec{y}_k - \vec{y} \quad (2.9)$$

Eine Möglichkeit das Vorzeichen des Netzfehlers weiterhin für die Berechnung der Gewichtsänderung nutzbar zu machen, ist die alternative Verwendung der Ableitung der betrachteten Netzausgabe nach den sie beeinflussenden Verbindungsgewichten $\partial y_j / \partial w_{ij}^{(l)}$ [66]. Diese Ableitungen werden in der sogenannten Jacobi-Matrix $\mathbf{J}(\mathbf{w})$ zusammengefasst. Dabei enthält eine Zeile der Jacobi-Matrix die Ableitungen nach den relevanten Gewichten je einer Ausgabegröße y_j . In Abbildung 2.9 ist zur Veranschaulichung ein Netzwerk mit zwei Ausgangsgrößen dargestellt.

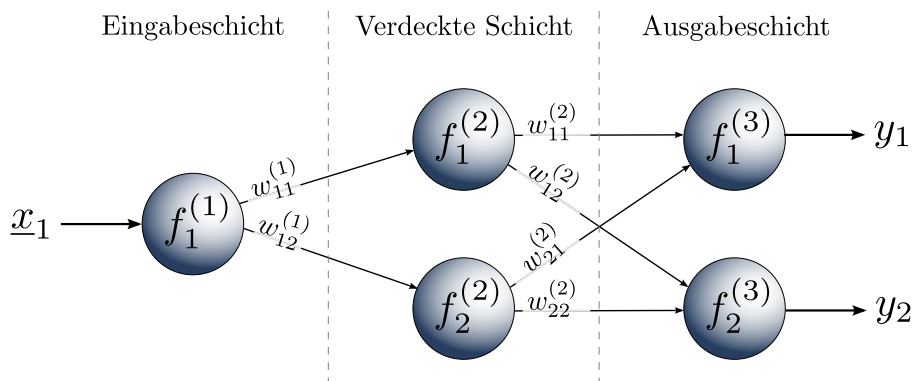


Abbildung 2.9: Beispielsnetzwerk mit zwei Ausgabeneuronen zur Veranschaulichung der Jacobi-Matrix Einträge

Es ist zu erkennen, dass die Gewichte $w_{12}^{(2)}$ und $w_{22}^{(2)}$ keine Beiträge zur Bildung der Netzausgabe y_1 beisteuern. Analog verhält es sich mit den Gewichten $w_{11}^{(2)}$ und $w_{21}^{(2)}$ für die Ausgabe

y_2 . In der Jacobi-Matrix werden solche Gewichte zu Null gesetzt. In Gleichung 2.10 ist die Jacobi-Matrix für das in Abbildung 2.9 gezeigte Beispielnetz gegeben.

$$\mathbf{J}(\mathbf{w}) = \begin{bmatrix} \frac{\partial y_1}{\partial w_{11}^{(1)}} & \frac{\partial y_1}{\partial w_{12}^{(1)}} & \frac{\partial y_1}{\partial w_{11}^{(2)}} & 0 & \frac{\partial y_1}{\partial w_{21}^{(2)}} & 0 \\ \frac{\partial y_2}{\partial w_{11}^{(1)}} & \frac{\partial y_2}{\partial w_{12}^{(1)}} & 0 & \frac{\partial y_2}{\partial w_{12}^{(2)}} & 0 & \frac{\partial y_2}{\partial w_{22}^{(2)}} \end{bmatrix} \quad (2.10)$$

Für ein neuronales Netzwerk mit nur einer Netzausgabe, wie es für die Untersuchungen in dieser Arbeit verwendet wird, kann die Jacobi-Matrix entsprechend zu einem Zeilenvektor reduziert werden. Anhand der Jacobi-Matrix wird auch anschaulich dargestellt, woher der Backpropagation-Algorithmus seinen Namen hat. So ist es nicht ohne weiteres möglich, den Einfluss eines Gewichts zwischen den vorderen Schichten auf den Ausgang zu quantifizieren. Um dies tun zu können, ist es nötig die m Netzwerkausgaben rückwärts durch das Netzwerk zu propagieren. Wo also die Berechnung der Ausgänge einer Propagation vorwärts durchs Netz entspricht, wird das Training durch eine umgekehrte, rückwärts gerichtete Propagation durchgeführt. Dafür ist es notwendig, die Einträge der Jacobi-Matrix umzuschreiben. Mit Hilfe der Kettenregel folgt nach [66] einzeln für jede Netzausgabe:

$$\frac{\partial y_m}{\partial w_{ij}^{(l)}} = \frac{\partial y_m}{\partial n_j^{(l+1)}} \cdot \frac{\partial n_j^{(l+1)}}{\partial w_{ij}^{(l)}}. \quad (2.11)$$

Hier kann die Neuroneneingabe $\partial n_j^{(l+1)}$ mit Gleichung 2.1 umgeschrieben werden zu:

$$\frac{\partial y_m}{\partial w_{ij}^{(l)}} = \frac{\partial y_m}{\partial n_j^{(l+1)}} \cdot \frac{\partial \left(\sum_{i=1}^q o_i^{(l)} \cdot w_{ij}^{(l)} \right)}{\partial w_{ij}^{(l)}} = \frac{\partial y_m}{\partial n_j^{(l+1)}} \cdot o_i^{(l)}. \quad (2.12)$$

An dieser Stelle kann der sogenannte rückpropagierte Ausgang $\hat{\delta}_j^{(l)}$ eingeführt werden, der die Empfindlichkeit der Netzausgabe in Abhängigkeit der Neuroneneingänge angibt.

$$\hat{\delta}_{j,m}^{(l)} = \frac{\partial y_m}{\partial n_j^{(l)}} \quad (2.13)$$

Mit Hilfe von $\hat{\delta}_j^{(l)}$ ist es nun möglich, rückwärts durch das Netz zu propagieren und schichtweise den Einfluss der Gewichte auf die Netzausgabe y_m zu berechnen. Angefangen mit der Ausgangsschicht, für die der Neuronenausgang $o_m^{(L)}$ gleich der Netzausgabe y_m ist, folgt für den m -ten rückpropagierten Ausgang:

$$\hat{\delta}_{j,m}^{(L)} = \frac{\partial y_m}{\partial n_m^{(L)}} = \frac{\partial o_m^{(L)}}{\partial n_m^{(L)}} = \mathbf{f}_m^{\prime(L)}. \quad (2.14)$$

Dabei wird die Änderung der Neuronenausgabe abhängig von der Änderung der Neuroneneingabe als Ableitung der Aktivierungsfunktion $\mathbf{f}_m^{\prime(L)}$ umgeschrieben, was die geforderte Differenzierbarkeit der Transferfunktion aus Abschnitt 2.2.1 begründet. Für die weitere

Rückpropagation in die vorderen Schichten des Netzwerks muss nun berücksichtigt werden, dass sich bei Netzwerken mit mehreren Ausgängen die Ausgaben der verdeckten Neuronen auf mehrere Ausgangsneuronen verteilen. Es ist somit notwendig, die Auswirkung der Änderung aller Neuronenausgaben $o_j^{(l)}$ auf die m Ausgänge zu berechnen. Mit Anwendung der Kettenregel folgt:

$$\widehat{\delta}_{j,m}^{(l)} = \frac{\partial y_m}{\partial n_j^{(l)}} = \frac{\partial y_m}{\partial o_j^{(l)}} \cdot \frac{\partial o_j^{(l)}}{\partial n_j^{(l)}}. \quad (2.15)$$

Es ist wichtig zu erwähnen, dass eine einzelne Neuronenausgabe $o_j^{(l)}$ nur einen indirekten Einfluss auf die Netzausgabe y_m hat, da sie zunächst noch mit allen weiteren Ausgaben der vorherigen Schicht zur konsolidierten Netzeingabe $n^{(l+1)}$ aufsummiert wird. Aus diesem Grund wird erneut die Kettenregel verwendet, um die Netzausgabe direkt nach allen Netzeingängen abzuleiten, um diese dann im Anschluss zu summieren. Dies entspricht der Bildung des totalen Differentials.

$$\widehat{\delta}_{j,m}^{(l)} = \frac{\partial y_m}{\partial n_j^{(l)}} = \sum_{i=1}^{q^{(l+1)}} \left(\frac{\partial y_m}{\partial n_i^{(l+1)}} \cdot \frac{\partial n_i^{(l+1)}}{\partial o_j^{(l)}} \cdot \frac{\partial o_j^{(l)}}{\partial n_j^{(l)}} \right) = \mathbf{f}_j^{\prime(l)} \cdot \sum_{i=1}^{q^{(l+1)}} \left(\widehat{\delta}_{i,m}^{(l+1)} \cdot w_{ij}^{(l)} \right). \quad (2.16)$$

Um zu dieser übersichtliche Darstellung mit den Ableitungen der Transferfunktionen und dem rückpropagierten Ausgang zu gelangen, wurden erneut die bereits oben vorgestellten Zusammenhänge

$$\frac{\partial o_j^{(l)}}{\partial n_j^{(l)}} = \mathbf{f}_j^{\prime(l)}, \quad \frac{\partial y_m}{\partial n_i^{(l+1)}} = \widehat{\delta}_{i,m}^{(l+1)} \quad \text{und} \quad n_i^{(l+1)} = \sum_{j=1}^{q^{(l)}} \left(o_j^{(l)} \cdot w_{ij}^{(l)} \right) \quad (2.17)$$

angewendet und eingesetzt. Mit Gleichung 2.16 ist nun eine Formel gefunden, mit deren Hilfe der rückpropagierende Ausgang einer Schicht in Abhängigkeit des rückpropagierten Ausgangs der anschließenden Schicht berechnet werden kann. Auf diese Weise kann dieser nun rückwärts durch das komplette Netzwerk propagiert und so die Jacobi-Matrix vervollständigt werden. In Matrixschreibweise können die einzelnen Zeilen der Jacobi-Matrix nach [66] wie folgt zusammengefasst werden:

$$\frac{\partial y_m}{\partial \mathbf{w}^{(l)}} = \begin{bmatrix} \frac{\partial y_m}{\partial w_{11}^{(l)}} & \cdots & \frac{\partial y_m}{\partial w_{1q^{(l+1)}}^{(l)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial w_{q^{(l)}1}^{(l)}} & \cdots & \frac{\partial y_m}{\partial w_{q^{(l)}q^{(l+1)}}^{(l)}} \end{bmatrix} \quad \text{und} \quad \widehat{\delta}_m^{(l)} = \begin{bmatrix} \widehat{\delta}_{1,m}^{(l)} \\ \vdots \\ \widehat{\delta}_{q^{(l)},m}^{(l)} \end{bmatrix}. \quad (2.18)$$

Somit ergibt sich in übersichtlicher Schreibweise die Berechnung der Einträge der Jacobi-Matrix einer Schicht l zu

$$\frac{\partial y_m}{\partial \mathbf{w}^{(l)}} = \left[\widehat{\delta}_m^{(l+1)} \cdot \mathbf{o}^{(l)T} \right]^T. \quad (2.19)$$

Mit der nun vollständig bekannten Jacobi-Matrix kann diese durch eine weitere Anwendung der Kettenregel in die Gewichtsänderung des Backpropagation-Algorithmus in Gleichung 2.8 eingesetzt werden. Mit dem Wissen, dass $\partial E / \partial \vec{y} = -(\vec{y}_k - \vec{y})$ gilt, ergibt sich für ∇E :

$$\frac{\partial E}{\partial \mathbf{w}} = \frac{\partial E}{\partial \vec{y}} \cdot \frac{\partial \vec{y}}{\partial \mathbf{w}} = - \left[\left(\frac{\partial \vec{y}}{\partial \mathbf{w}} \right)^T \cdot (\vec{y}_k - \vec{y}) \right] = -\mathbf{J}^T \cdot \vec{\epsilon}. \quad (2.20)$$

Durch Einsetzen von Gleichung 2.20 in Gleichung 2.8 ergibt sich abschließend die finale Vorschrift zur Änderung der Verbindungsgewichte für den Backpropagation-Algorithmus mit Verwendung der Jacobi-Matrix.

$$\Delta \mathbf{w} = -\mu \cdot \frac{\partial E}{\partial \mathbf{w}} = \mu \cdot \mathbf{J}^T \cdot \vec{\epsilon} \quad (2.21)$$

Für die in Kapitel 5 vorgestellten Erweiterungen der Gewichtsänderung als Mittel zur Stabilisierung des geschlossenen Regelkreises auf Basis dynamischer Inversion mit neuronalen Netzwerken in der innersten Kaskade, kann Gleichung 2.21 aufgeteilt werden in $\Delta \mathbf{w}^{(1)}$ und $\Delta \mathbf{w}^{(2)}$. Somit folgt für ein dreischichtiges neuronales Netzwerk durch Einsetzen der oben dargestellten Zusammenhänge direkt:

$$\Delta \mathbf{w}^{(1)} = \mu^{(1)} \cdot \left(\frac{\partial \vec{y}}{\partial \mathbf{w}^{(1)}} \right)^T \cdot \vec{\epsilon} = \mu^{(1)} \cdot \left[\hat{\delta}_m^{(2)} \cdot \mathbf{o}^{(1)T} \right]^T \cdot \vec{\epsilon} = \mu^{(1)} \cdot \left[\text{diag}(\mathbf{f}'^{(2)}) \cdot \mathbf{w}^{(2)} \cdot \underline{\vec{x}} \right]^T \cdot \vec{\epsilon}, \quad (2.22)$$

$$\Delta \mathbf{w}^{(2)} = \mu^{(2)} \cdot \left(\frac{\partial \vec{y}}{\partial \mathbf{w}^{(2)}} \right)^T \cdot \vec{\epsilon} = \mu^{(2)} \cdot \left[\hat{\delta}_m^{(3)} \cdot \mathbf{o}^{(2)T} \right]^T \cdot \vec{\epsilon} = \mu^{(2)} \cdot \left[f^{(2)}(\mathbf{w}^{(1)} \cdot \underline{\vec{x}}) \right]^T \cdot \vec{\epsilon}. \quad (2.23)$$

Dabei handelt es sich bei $\mu^{(1)}$ und $\mu^{(2)}$ um die einzelnen Lernraten der beiden Gewichtsmatrizen $\mathbf{w}^{(1)}$ und $\mathbf{w}^{(2)}$. Durch das Gleitzustandstraining aus Kapitel 4 vorgegeben, wird innerhalb dieser Arbeit jedoch nur ein einzelnes μ für das gesamte Netzwerk berechnet, so dass zu jeder Zeit $\mu^{(1)} = \mu^{(2)}$ gilt.

Das Ziel der Gewichtsänderung ist es, in möglichst kurzer Zeit die optimale Gewichtsconfiguration $\mathbf{w}_*^{(1)}$ und $\mathbf{w}_*^{(2)}$ zu etablieren, bei der der verbleibende Approximationsfehler $\vec{\epsilon}_A$ minimal ist [48]. Dabei existiert für den Approximationsfehler eine obere Grenze $\|\vec{\epsilon}_A\| < \bar{\epsilon}$, die mit steigender Anzahl der Neuronen in der verdeckten Schicht sinkt. Mit $\vec{\epsilon}_A$ und den optimalen Gewichtsmatrizen lässt sich nach [7,8] die Approximation jeder beliebigen nichtlinearen Funktion durch ein Netzwerk mit geeignet vielen Neuronen in der verdeckten Schicht gewährleisten. Für die Abbildung einer glatten Funktion $f(\vec{x}) : \mathfrak{R}^n \rightarrow \mathfrak{R}^m$ lässt sich so für jedes kompakte Gebiet $D \subset \mathfrak{R}^n$ die Approximation durch ein neuronales Netzwerk wie folgt formulieren:

$$f(\vec{x}) = \mathbf{w}_*^{(2)} \cdot \vec{f}^{(2)}(\mathbf{w}_*^{(1)} \cdot \underline{\vec{x}}) + \vec{\epsilon}_A \quad \text{mit} \quad \forall \vec{x} \in D \subset \mathfrak{R}^n. \quad (2.24)$$

2.3.2 Herausforderungen der Nutzung Online-trainierter neuronaler Netzwerke in Regelungsanwendungen

Die dargestellte Adaptionfähigkeit und die somit gegebene Möglichkeit, sich unbekannten nichtlinearen Umgebungen anzupassen, macht Online-trainierte neuronale Netzwerke zu leistungsfähigen Werkzeugen in regelungstechnischen Anwendungen. Dabei ist jedoch das oftmals angeführte Black-Box-Verhalten [2,67,68] problematisch, insbesondere wenn es sich um den Einsatz in sicherheitsrelevanten Systemen wie beispielsweise der Regelung von Straßen- und Luftfahrzeugen handelt. Für Systeme, bei denen ein technisches Versagen, sei es auf Software- oder Hardware-Ebene, zum Verlust von Leben führen kann, gelten strenge Regularien, deren Einhaltung in dezidierten Zertifizierungsprozessen nachgewiesen werden muss. Durch das charakteristische Verhalten von adaptiven Regelungsverfahren, sich über den gesamten Betriebszustand kontinuierlich zu verändern und anzupassen, ist die Anwendbarkeit konventioneller Zulassungsverfahren nicht gegeben oder zumindest nur beschränkt möglich. Nach [67] sind dynamisch variierende Regelungsstrecken, die gegebene Nichtlinearität neuronaler Netzwerke und nicht zuletzt die Komplexität und die wahrgenommene Intransparenz der verwendeten Lernverfahren Motivation für die Entwicklung neuer Validierungs- und Verifikationsmethoden (V&V).

Um sich der zertifizierten Anwendung von neuronalen Netzen und neuro-adaptiven Reglern in sicherheitskritischen Systemen zu nähern, werden in der Literatur verschiedene V&V Ansätze untersucht und diskutiert [53,67,69]. Diese können nach [67] in den unten stehenden Kategorien zusammengefasst werden:

- Analyse von Besonderheiten und Charakteristiken der Trainingsalgorithmen
- Erzeugung von Transparenz der inneren Abläufe neuronaler Netze
- Getrennte V&V Ansätze für die Programmierung nach Qualitätsmanagementrichtlinien und die theoretische Stabilitätsanalyse der Algorithmen (bspw. Ljapunow)
- Betrachtung von neuronalen Netzwerken als Funktionsapproximatoren und Klassifikationswerkzeuge
- Analyse der Auswahl und Qualität der im Netzwerk genutzten Daten
- Ansätze zur dynamischen Überwachung der adaptiven Komponenten

Die in dieser Arbeit vorgestellten Konzepte und Methoden des Gleitzustandtrainings fallen dabei in die oberen beiden Kategorien. So wird in den folgenden Kapiteln der klare Fokus auf die Untersuchung von Konvergenzgeschwindigkeit und Stabilität der vorgeschlagen Trainingsalgorithmen gelegt. Ein wesentlicher Aspekt ist dabei die Untersuchung der automatischen und dynamisch für jeden Zeitschritt wiederholten Berechnung einer stabilen Lernrate. Wie bei Gradientenabstiegsverfahren üblich, kann eine falsch eingestellte Schrittweite zu schlechtem Konvergenzverhalten oder Oszillationen führen, die besonders

im Falle eines direkten Einflusses des Netzwerks auf mögliche Steuergrößen dringend zu vermeiden sind. Aus diesem Grund ist ein wesentlicher Aspekt, der in dieser Arbeit vorgestellten Lernverfahren, die Aufstellung stabiler Grenzen innerhalb derer eine geeignete Lernrate ermittelt wird. Auf diese Weise entfällt der empirische Auslegungsprozess der Lernrate, der unabdingbar mit einer großen Anforderung an Zeit und Erfahrung verknüpft ist, weg und verspricht eine steigende Akzeptanz und eine beschleunigte Verwendung von neuronalen Netzwerken in sicherheitskritischen Anwendungen.

Darüber hinaus wird das aus der strukturvariablen Regelung abgeleitete Gleitzustandslernen dafür verwendet, die internen Prozesse eines neuronalen Netzes und dessen Zustandsänderung transparenter zu gestalten, indem durch den Einsatz sogenannter Schaltgesetze der Netzwerkfehler \vec{e} auf eine vorgegebene Trajektorie im Zustandsraum gezwungen wird. So kann ein mit klassischen Trainingsverfahren nicht zu realisierender Grad an Determinismus bezüglich der Änderung des Netzwerkfehlers im kommenden Zeitschritt erreicht werden. Darauf aufbauend wird in einer theoretischen Stabilitätsbetrachtung nach [46] und [48] die Stabilität der verwendeten Algorithmen im geschlossenen Regelungssystem untersucht und in anschließenden Simulationen getestet. Auf diese Weise soll ein Beitrag zur zukünftigen Anwendung von Online-trainierten neuronalen Netzwerken in adaptiven Regelungssystemen geleistet werden.

Kapitel 3

Strukturvariable Regelung

Das Prinzip der strukturvariablen Regelung (engl. variable structure system - VSS), wenn auch noch nicht als solche bezeichnet, wurde erstmals Ende der 1950er Jahre von Stanislaw Emelyanov vorgestellt [70] und in den folgenden Jahren und Jahrzehnten aufgrund der für den Ansatz charakteristischen Robustheit gegenüber Parameterunsicherheiten und Störungen umfassend praktisch eingesetzt und weiterentwickelt [71–73]. Die dem Ansatz zugrunde liegende Idee ist es, einen Regler zu entwerfen, der durch die Nutzung sogenannter Schaltgesetze zwischen verschiedenen parallelen Teilsystemen umschalten kann. Auf diese Weise ist es dem Regler möglich, seine Struktur zu verändern, was von der einfachen Selektion verschiedener Parametersätze bis hin zu der Umschaltung zwischen gänzlich verschiedenen Übertragungsverhalten reicht. In Abbildung 3.1 ist eine vereinfachte und abstrakte Abbildung eines strukturvariablen Reglers mit der Möglichkeit zur Umschaltung zwischen zwei unterschiedlichen Strukturen dargestellt.

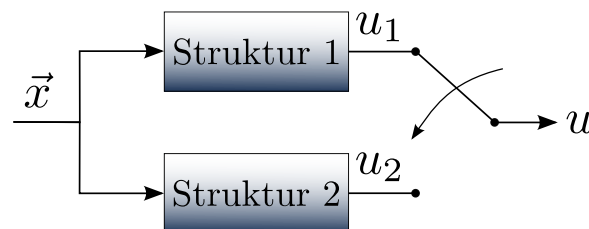


Abbildung 3.1: Abstrakte Darstellung eines strukturvariablen Systems.

In Abhängigkeit des verwendeten Schaltgesetzes und des zurückgeführten Zustandsvektors \vec{x} wird in diesem Beispiel zwischen den strukturabhängigen Steuergrößen u_1 und u_2 umgeschaltet. Als Kriterium für das Schaltgesetz, wann umgeschaltet werden soll, wird oftmals eine sogenannte zustandsabhängige Schaltfunktion $S(\vec{x})$ oder in Vektorschreibweise ein Schaltfunktionsvektor $\vec{S}(\vec{x}) = [S_1(\vec{x}), S_2(\vec{x}), \dots, S_m(\vec{x})]^T$ mit m Einträgen verwendet. Es ist auch möglich, ohne eine Schaltfunktion direkt an einzelnen Systemzuständen, in diesem Zusammenhang als Constraint bezeichnet, zu schalten. Jedoch erlaubt die Verwendung einer Schaltfunktion eine präzisere Beschreibung von Zustandskombinationen zur Erzeugung eines exakten Schaltregimes im Phasenraum.

Ein diskontinuierliches Schaltgesetz für eine beispielhafte strukturvariable Regelung, wie sie in Abbildung 3.1 dargestellt ist, kann somit wie folgt aussehen:

$$u(\vec{x}) = \begin{cases} u_1(\vec{x}) & \text{für } S(\vec{x}) > 0 \\ u_2(\vec{x}) & \text{für } S(\vec{x}) < 0 . \end{cases} \quad (3.1)$$

Es sei angemerkt, dass bei der Bezeichnung des Schaltregimes die Ordnung des Systems berücksichtigt wird. Handelt es sich bei dem Schaltregime um eine Linie im Zustandsraum, wird von einer Schaltlinie gesprochen. Wird das Schaltregime durch eine Fläche definiert, verändert sich der Ausdruck entsprechend zu Schaltfläche und bei weiterer Erhöhung der Systemordnung zu Schalthyperfläche. Existieren mehrere Schaltfunktionen und entsprechend unterschiedliche Schaltregime innerhalb eines Systems, können diese zusammengefasst auch als Schaltmannigfaltigkeit bezeichnet werden. Innerhalb dieser Arbeit wird, bedingt durch die in späteren Kapiteln dargestellten Anwendungen, überwiegend von Schaltlinien oder Schaltregimen gesprochen.

Ein wesentlicher Aspekt bei der Nutzung eines strukturvariablen Reglers ist die Erzeugung neuer Systemeigenschaften, welche keiner der einzelnen Substrukturen für sich genommen zu eigen sind. Dabei ist es nach [72] klar, dass in Abhängigkeit des Schaltgesetzes und der einzelnen Strukturen sowohl stabile als auch instabile Gesamtsysteme entstehen können. Ein oftmals vorgestelltes Beispiel ist dabei die Kombination zweier grenzstabiler Systeme, wie sie in Abbildung 3.2 dargestellt sind, zu einem stabilen Gesamtsystem durch die Formulierung eines geeigneten Schaltgesetzes [74].

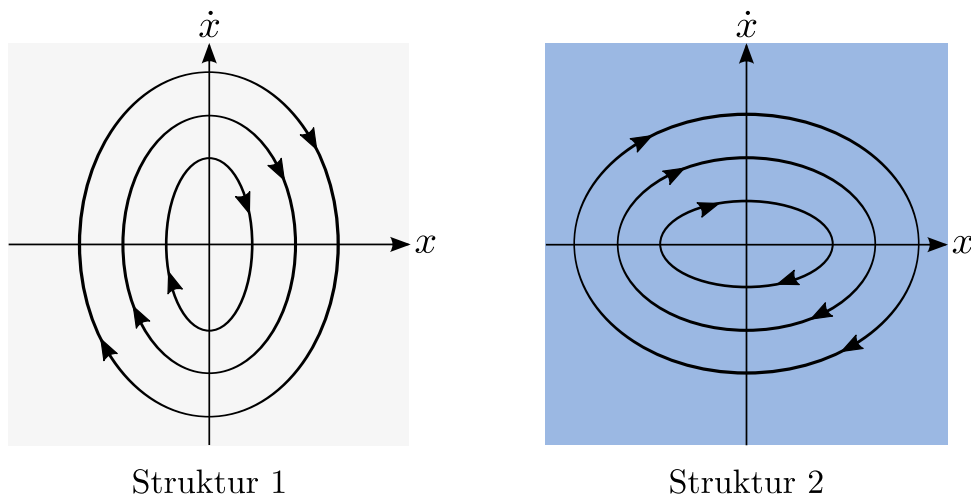


Abbildung 3.2: Darstellung zweier grenzstabiler Strukturen in der Phasenebene

Durch ein diskontinuierliches Schaltgesetz, dass jeweils an den Koordinatenachsen zwischen beiden Strukturen umschaltet, lässt sich ein asymptotisch stabiles Gesamtsystem erzeugen, dessen prinzipielles Verhalten in der Phasenebene in Abbildung 3.3 dargestellt ist.

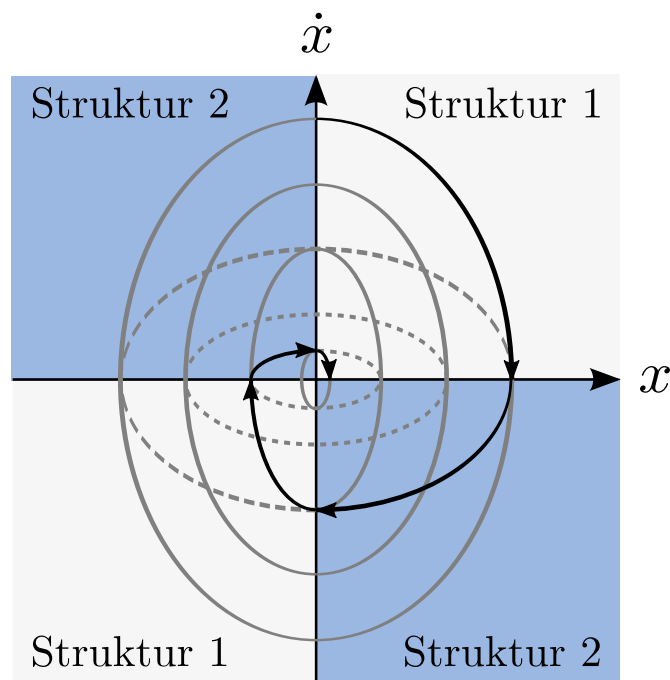


Abbildung 3.3: Darstellung der kombinierten Systemdynamik von Struktur 1 und 2 nach [74].

Beginnend im oberen rechten Quadranten verläuft das System entsprechend dem in Abbildung 3.2 gezeigten Verhalten von Struktur 1 entlang einer aufrecht stehenden Ellipse, bis es auf die Abszissenachse trifft. An dieser Stelle erfolgt die Umschaltung auf Struktur 2 und somit der Wechsel zu der Zustandstrajektorie einer liegenden Ellipse nach Abbildung 3.2. Dabei ist das System durch das Verlassen der ursprünglichen Trajektorie von Struktur 1 dem Koordinatenursprung des Phasenraums bereits näher gekommen. An der Ordinatenachse erfolgt die erneute Umschaltung zurück zu Struktur 1. Auf diese Weise folgen weitere Umschaltungen, bis das System letztendlich im Ruhezustand bei $x = \dot{x} = 0$ angekommen ist. Es kann somit anschaulich demonstriert werden, wie mit Hilfe der strukturvariablen Regelung und der daraus resultierenden Umschaltung zwischen grenzstabilen Systemen ein asymptotisch stabiles Gesamtsystem erzeugt werden kann.

Neben dieser Kombination unterschiedlicher Struktureigenschaften wurde die strukturvariable Regelung in den 1960er Jahren dafür verwendet, gewünschte Zustandstrajektorien eines Systems direkt anzusteuern, indem ein Schaltgesetz entwickelt wird, dass das System startpunktunabhängig an einer dieser vorteilhaften Trajektorien entlang führt. Die vorgestellten Anwendungen von VSS sind nach [74] jedoch in ihrer Anwendbarkeit auf sehr spezifische Problemstellung niedriger Ordnung beschränkt und konnten den, im frühen Stadium der Forschung entstandenen Erwartungen, nicht gerecht werden. Den schlussendlichen Durchbruch als leistungsstarkes und speziell für unbekannte nichtlineare Systeme verwendbares Verfahren hatte die strukturvariable Regelung durch eine besondere Form ihrer Anwendung als sogenannte Gleitzustandsregelung (engl. Sliding-Mode-Control - SMC), wie sie im folgenden Abschnitt eingeführt wird.

3.1 Gleitzustandsregelung

Bei der Gleitzustandsregelung handelt es sich um die Erzeugung eines speziellen Systemzustands, dem Gleitzustand, bei dem das Schaltgesetz zu einer Umschaltung in unendlich hoher Frequenz zwischen verschiedenen Systemstrukturen führt. Auf diese Weise nähern sich die Systemzustände dem gewählten Schaltregime von unterschiedlichen Seiten an, bis sie dieses letztendlich erreichen und durch das Schaltgesetz forciert auch nicht mehr verlassen. Entlang der Schaltlinie, nun folgerichtig als Gleitlinie bezeichnet, gleiten die Zustände anschließend in den Ruhezustand. Bei der Verwendung eines Schaltgesetzes wie es in Gleichung 3.1 dargestellt ist, mit dem diskontinuierlichem Umschalten beim Wechsel des Vorzeichens der Schaltfunktion, kann definiert werden, dass die Zustände die Schaltlinie erreicht haben, wenn $S = 0$ gilt. Wenn darüber hinaus noch $\dot{S} = 0$ gilt, handelt es sich um die Gleitlinie, an der sich der robuste Gleitzustand einstellt. Veranschaulicht wird das Eintreffen und Gleiten entlang der Gleitlinie in Abbildung 3.4.

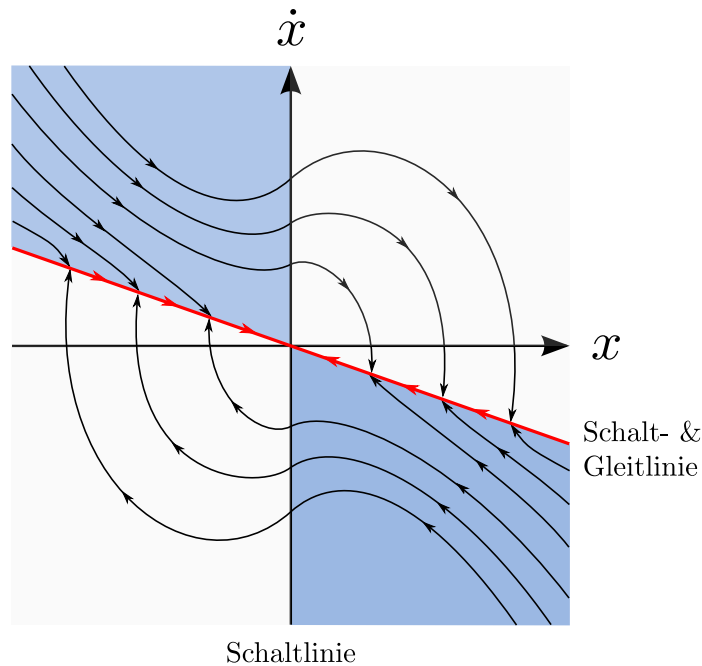


Abbildung 3.4: Strukturvariable Regelung im Gleitzustand.

Dargestellt ist ein System mit einer senkrechten Schaltlinie bei $x = 0$ und einer diagonal verlaufenden Gleitlinie mit der Geradengleichung $\dot{x} = -0,5 \cdot x$ welche beide aus der Schaltfunktion $S(x, \dot{x}) = x(\dot{x} + 0,5 \cdot x) = 0$ abgeleitet sind. Es ist deutlich zu erkennen, wie die Zustandstrajektorien von beiden Seiten auf die Gleitlinie treffen, dort durch einen theoretisch unendlich schnellen Schaltvorgang gehalten werden und in Folge dessen in den Ursprung gleiten. So kann von einem Gleitzustand an einem diskontinuierlichen Regime gesprochen werden, wenn S und dessen Änderung \dot{S} in der unmittelbaren Nähe des Gleitregimes unterschiedliche Vorzeichen aufweisen [72]:

$$\lim_{S \rightarrow 0^-} \dot{S} > 0 \quad \text{und} \quad \lim_{S \rightarrow 0^+} \dot{S} < 0. \quad (3.2)$$

Charakteristisch für die Nutzung eines Gleitzustandsreglers ist die Veränderung der geschlossenen Systemdynamik nach Erreichen des Gleitzustands. Wie in Abbildung 3.4 zu sehen ist, ändert sich die gezeigte Dynamik einer klassischen Strudel- und Sattelpunkttrajektorie zu einer linearen Bewegung durch den Phasenraum. Es kann somit festgehalten werden, dass das geregelte System nach Eintreffen auf der Gleitlinie einer durch den Gleitzustand vorgegebenen Dynamik folgt und diese Dynamik eine reduzierte Ordnung gegenüber der ursprünglichen Regelstrecke besitzt [75]. Zur detaillierteren Veranschaulichung der Dynamikänderung des geschlossenen Gesamtsystems wird im Anschluss ein kurzes Beispiel zur Vorgabe der Zustandstrajektorie durch die Wahl der Schaltfunktion in Anlehnung an [76] gegeben.

Beispiel: Zustandstrajektorie im Gleitzustand

Gegeben sei ein unbekanntes dynamisches System zweiter Ordnung

$$\ddot{x} = a(t) + b(t) \cdot u \quad (3.3)$$

mit den Unbekannten

$$|a| \leq 1, \quad 1 \leq b \leq 2 \quad (3.4)$$

und Affinität in der Steuergröße $u \in \Re$. Das System soll im Ursprung stabilisiert werden, so dass eine Schaltfunktion S wie folgt definiert wird:

$$S = \dot{x} + \lambda \cdot x. \quad (3.5)$$

Dabei ist λ der Parameter, der die Steigung der resultierenden Schaltlinie im Phasenraum angibt. Er beschreibt das Verhältnis des Fehlers und seiner Änderungsrate nach Erreichen des Gleitzustands. Ein Schaltgesetz zur lokalen Erreichung des Schaltregimes $S = 0$ folgt zu:

$$u = -2 \cdot \text{sign}(S). \quad (3.6)$$

In Abbildung 3.5 sind die Zustandstrajektorien des gleitzustandsgeregelten Systems zweiter Ordnung mit den initialen Werten von $x(0) = 0$ und $\dot{x}(0) = 1$, sowie $b = 2$ und $a = 0,5$ bzw. $a = 1$ dargestellt. Anhand Abbildung 3.5 - (a) ist gut zu erkennen, dass das geregelte System nach Erreichen des Gleitzustands eine geänderte Dynamik annimmt und diese durch die Schaltfunktion vorgegeben werden kann. So lässt sich durch die Wahl des Parameters λ beispielsweise die Steigung der Trajektorie im Phasenraum vorgeben. Ebenfalls dargestellt ist die Veränderung der Schaltfunktion in Form der Berücksichtigung der Wurzelfunktion. Wie anzunehmen, besitzt das System in Folge dessen nach Erreichen des Gleitzustands eine nichtlineare Zustandstrajektorie.

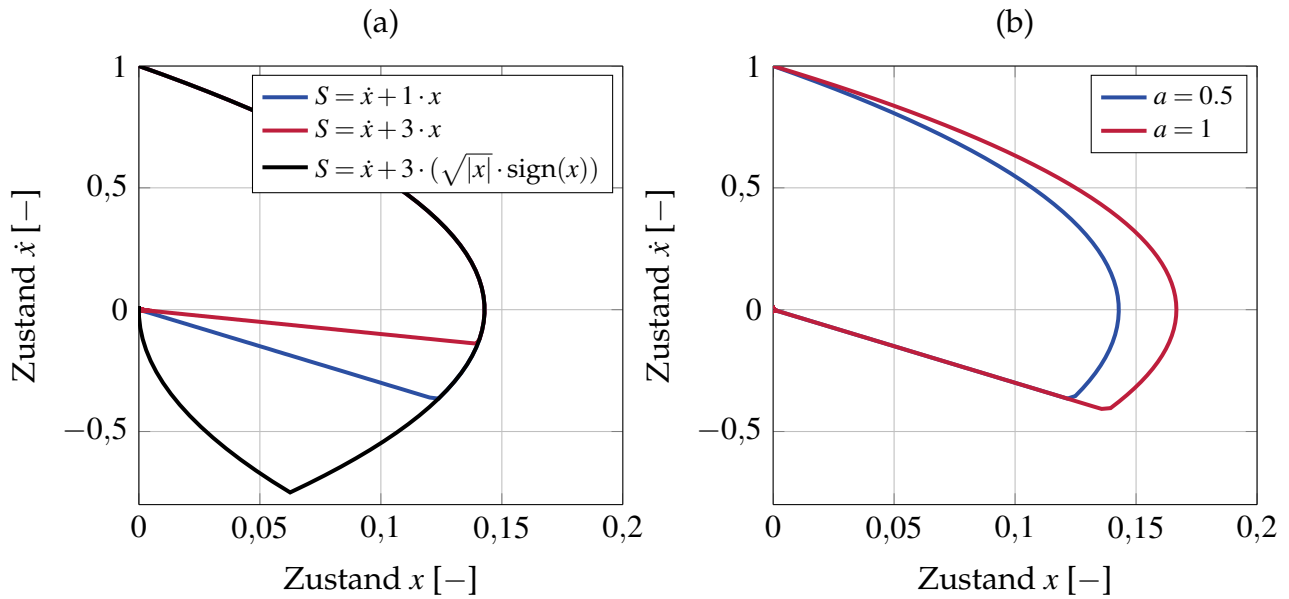


Abbildung 3.5: Beispielhafte Darstellung der Zustandstrajektorien im Phasenraum für (a) unterschiedliche Schaltfunktionen und (b) unterschiedliche Systemeigenschaften

Ein weiteres Beispiel für die Unabhängigkeit des Gleitzustands von der ursprünglichen Systemdynamik ist in Abbildung 3.5 - (b) dargestellt. Trotz unterschiedlicher Systemeigenschaften und somit prinzipiell unterschiedlichen Trajektorien in der Phasebene verhalten sich beide Systeme identisch, sobald sie die Gleitlinie erreicht haben. Durch den Einsatz der Gleitzustandsregelung kann somit unterschiedlichen Systemen die gleiche Dynamik aufgeprägt werden. Dabei besitzt das geregelte System im Gleitzustand eine erhöhte Robustheit gegenüber Störungen und unbekannten Systemparametern [74, 76, 77]. Die in diesem Beispiel dargestellte stabile Konvergenz der Zustände in den Ursprung ist jedoch nicht allgemeingültig auf alle Gleitzustandsverfahren zu übertragen. Die Stabilitätsbetrachtung komplexerer strukturvariabler Systeme stellt eine herausfordernde Aufgabe dar und kann als Untersuchung der Nulldynamik (vgl. Kapitel 5) der Schaltfunktion S verstanden werden [75].

3.1.1 Verhalten im Gleitzustand

Die Besonderheit bei der Betrachtung der Stabilität und Störanfälligkeit einer Gleitzustandsregelung ist das theoretisch unendlich schnelle Umschalten der Struktur durch eine intrinsisch diskontinuierliche Steuergröße. Dieses für eine Regelung unübliche Verhalten führt dazu, dass weder von mathematischer noch von praktischer Seite die Dynamik eines komplexen Systems im Gleitzustand klar formuliert werden kann. Das Schaltgesetz deckt üblicherweise nur die Fälle $S < 0$ und $S > 0$ ab, was zur Folge hat, dass das Verhalten im Gleitzustand selber nicht definiert ist. Dies führt zu dem mathematischen Problem der Lösung von Differentialgleichungen mit diskontinuierlicher rechter Seite. Eine Möglichkeit zur Be-

stimmung einer Lösung ist die geometrische Herangehensweise nach Filipov [78]. Demnach kann für ein dynamisches System

$$\dot{\vec{x}} = f(\vec{x}, t, \vec{u}) \quad \text{mit} \quad \vec{x} \in \mathbb{R}^n \quad \text{und} \quad \vec{u} \in \mathbb{R}^m \quad (3.7)$$

mit einem diskontinuierlichen Stellgesetz

$$u_i = \begin{cases} u_i^+(\vec{x}, t) & \text{für } S_i(\vec{x}, t) > 0 \\ u_i^-(\vec{x}, t) & \text{für } S_i(\vec{x}, t) < 0 \end{cases} \quad i = 1, 2, \dots, m, \quad (3.8)$$

eine Lösung auf dem diskontinuierlichen Regime $S = 0$ gefunden werden, wenn diese die differentielle Inklusion

$$\dot{\vec{x}} \in V(\vec{x}, t) \quad (3.9)$$

erfüllt. Dabei entspricht der Vektorraum $V(\vec{x}, t)$ der konvexen Hülle aller Werte des Systems $f(\vec{x}, t, \vec{u})$ solange \vec{x} die vollständige Nachbarschaft δ von $S = 0$ abdeckt. So definiert, beschreibt der Geschwindigkeitsvektor $f_0(\vec{x}, t)$, als Teil des Vektorraums V , die Gleitzustandsdynamik als Tangente am diskontinuierlichen Regime. Diese Lösung wird üblicherweise als *Lösung nach Filipov* oder *Filipov'sche Lösung* bezeichnet. Eine abstrakte Darstellung dieses Ansatzes ist in Abbildung 3.6 zu sehen. Aufgrund ihrer geometrischen Natur ist die Methode nach Filipov nur sehr begrenzt auf praktische Systeme anzuwenden und dient somit primär zur Verdeutlichung, wie sich einer Dynamikbeschreibung des Gleitzustands durch Betrachtung der unmittelbaren Nachbarschaft des Gleitregimes genähert werden kann.

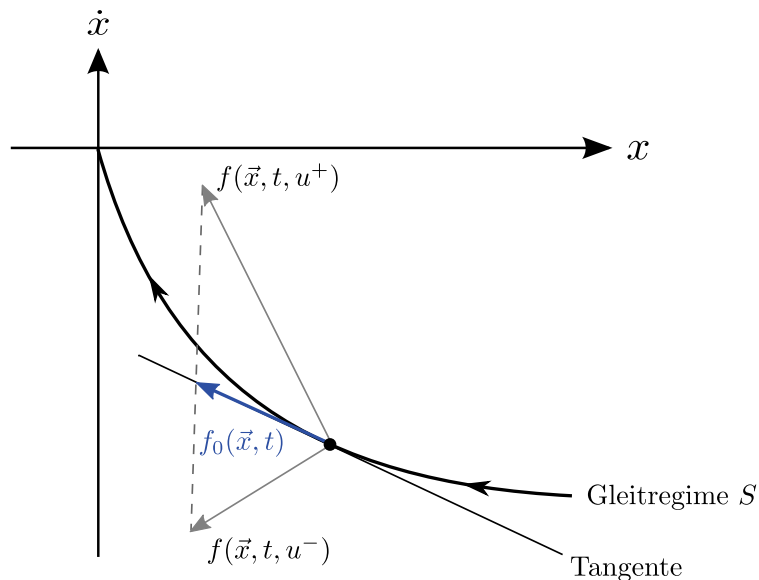


Abbildung 3.6: Geometrische Lösung der Systemdynamik f_0 im Gleitzustand nach Filipov

Eine weitere Möglichkeit zur Lösung von Differentialgleichungen mit diskontinuierlicher rechter Seite ist die äquivalente Regelung (engl. Equivalent-Control) nach Utkin [72]. Die

Methode basiert auf der Idee eine äquivalente Steuergröße u_{eq} zu finden, die für das Gleiten auf dem Regime $S = 0$ eindeutig definiert ist. Dies kann erreicht werden, indem die Gleichung $\dot{S} = 0$, als zwingende Bedingung für das Verbleiben im Gleitzustand, aufgestellt und nach der Steuergröße aufgelöst wird. Die so gefundene äquivalente Steuergröße entspricht einer Ersatzsteuergröße, die das geregelte System im Gleitzustand hält und kann somit für die Formulierung der Dynamik auf dem Gleitregime genutzt werden. Zur Verdeutlichung wird die äquivalente Regelung auf das oben gegebene Beispiel einer Gleitzustandsregelung eines Systems zweiter Ordnung 3.3 - 3.6 angewendet.

Beispiel: Äquivalente Regelung

Wie beschrieben wird für die Berechnung der äquivalenten Regelung die Ableitung der Schaltfunktion \dot{S} aufgestellt und nach der Steuergröße aufgelöst:

$$\dot{S} = \ddot{x} + \lambda \cdot \dot{x} = a(t) + b(t) \cdot u + \lambda \cdot \dot{x} = 0 \quad \Leftrightarrow \quad u_{eq} = \frac{-a(t) - \lambda \cdot \dot{x}}{b(t)} . \quad (3.10)$$

Setzt man die gefundene äquivalente Steuergröße nun in die Systembeschreibung aus Gleichung 3.2 für u ein, so erhält man:

$$\ddot{x} = a(t) + b(t) \cdot \frac{-a(t) - \lambda \cdot \dot{x}}{b(t)} = -\lambda \cdot \dot{x} \quad \text{mit} \quad \lambda > 0 . \quad (3.11)$$

Für das dargestellte System wurde somit eine Ersatzsteuergröße gefunden, die zu einem stabilen Verhalten im Gleitzustand bei $S = \dot{S} = 0$ führt, da nach den Gleichungen 3.11 und 3.5 die zeitliche Ableitung der Systemzustände stets von entgegengesetztem Vorzeichen sind und somit über die Zeit abklingen.

Durch die Verwendung der *Filipov'schen*, respektive *Utkin'schen Lösung* von Differentialgleichungen und -inklusionen mit diskontinuierlicher rechter Seite, ist es somit möglich Aussagen über die Stabilität von Gleitzustandsverfahren zu treffen. Für detaillierte Abhandlungen über Ljapunow-Stabilität, einfache, asymptotische und globale Stabilität strukturvariabler Systeme sowie dem Vergleich der Ansätze von Filipov und Utkin sei unter anderem auf [72,79–81] verwiesen. Im folgenden Abschnitt 3.1.2 soll sich im Vergleich dazu der Übersichtlichkeit halber auf die Darstellung der, für stabiles Verhalten zwingend notwendigen, Erreichbarkeits- und Existenzbedingungen eines Gleitzustands beschränkt werden.

3.1.2 Erreichbarkeit und Existenz des Gleitzustands

Für sowohl die Auslegung einer Gleitzustandsregelung als auch des im Kapitel 4 dargestellten Gleitzustandstrainings ist es unerlässlich Bedingungen aufzustellen, unter deren Einhaltung sowohl die Existenz als auch die Erreichbarkeit und Beibehaltung des Gleitzustands garantiert werden können. Zu diesem Zweck sind in der Literatur verschiedene Ansätze definiert, wie der Übergang von der Eintreffphase (engl. Reaching-Mode) in den

Gleitzustand sichergestellt werden kann [82–85]. Im Folgenden sollen drei in der Literatur häufig verwendete Methoden kurz dargestellt werden.

Direkter Ansatz der Schaltfunktion

Bei diesem Ansatz wird, ähnlich der Definition des Gleitzustands nach Utkin [72] aus Gleichung 3.2, das Erreichen des Gleitzustands durch die Unterschiedlichkeit der Vorzeichen von Schaltfunktion S und deren Änderung \dot{S} sicher gestellt. Es folgt somit, dass für alle m Schaltfunktionen des geregelten Systems

$$\dot{S}_i > 0 \text{ wenn } S_i < 0 \quad \text{und} \quad \dot{S}_i < 0 \text{ wenn } S_i > 0 \quad \forall i = 1, \dots, m \quad (3.12)$$

gelten muss. Anders ausgedrückt muss das Produkt aus S_i und \dot{S}_i stets negativ sein, so dass der Ausdruck umformuliert werden kann zu:

$$S_i \cdot \dot{S}_i < 0 \quad \forall i = 1, \dots, m. \quad (3.13)$$

Dabei handelt es sich um einen global gültigen Ausdruck, der, wie in Abschnitt 4.3 beschrieben, für zeitdiskrete Systeme jedoch nur ein notwendiges Kriterium darstellt und somit weder das Erreichen des Gleitzustands in finiter Zeit garantiert, noch das Auftreten von Instabilitäten bei zu großen Schrittweiten T_s verhindert.

Ljapunow-Ansatz

Durch die Wahl der Ljapunow-Funktion

$$V(\vec{x}, t) = \vec{S}^T \cdot \vec{S} \quad (3.14)$$

ist durch die negative Zeitableitung dieser Funktion

$$\dot{V}(\vec{x}, t) < 0 \quad \text{für } S \neq 0 \quad (3.15)$$

eine globale Erreichbarkeitsbedingung für den Gleitzustand gegeben [86]. Um finite Erreichbarkeit zu garantieren, kann Gleichung 3.15 variiert werden, indem verlangt wird, dass die negative Ableitung der Ljapunow-Funktion eine Obergrenze ε nicht überschreitet. Damit folgt:

$$\dot{V}(\vec{x}, t) < -\varepsilon \quad \text{für } \vec{S} \neq 0 \quad \text{mit } \varepsilon > 0. \quad (3.16)$$

Auf diese Weise werden kleine und gegen Null konvergierende Änderungsraten vor der finalen Erreichung des Gleitzustands verhindert.

Reaching-Law-Ansatz

Anders als bei den beiden vorherigen Bedingungen wird bei der Methode des Reaching-Laws gezielt das Systemverhalten in der Eintrittsphase beeinflusst. Beispielhaft kann ein Reaching-Law als erste zeitliche Ableitung der Schaltfunktion wie folgt allgemein definiert werden:

$$\dot{\vec{S}} = -\mathbf{Q} \cdot \text{sign}(\vec{S}) - \mathbf{K} \cdot \vec{h}(\vec{S}). \quad (3.17)$$

Dabei besitzen die Verstärkungsmatrizen \mathbf{Q} und \mathbf{K} ausschließlich positive Einträge auf den Hauptdiagonalen und die Funktion $\vec{h}(\vec{S})$ setzt sich aus m skalaren Funktion $h_i(S_i)$ zusammen

$$\vec{h}(\vec{S}) = [h_1(S_1), \dots, h_m(S_m)]^T, \quad (3.18)$$

welche die Bedingung

$$S_i \cdot h_i(S_i) > 0 \quad \text{für } S_i \neq 0, i = 1, \dots, m \quad (3.19)$$

erfüllen. Durch die Veränderung der Verstärkungsmatrizen, sowie der Struktur des Reaching-Laws kann die Dynamik der Eintrittsphase modifiziert werden. Bei drei häufig angeführten Beispielen handelt es sich um leicht abgewandelte Varianten der allgemeinen Formulierung aus 3.17. Der Übersichtlichkeit halber für skalare Schaltfunktionsänderungen formuliert, folgt:

1. Konstante Änderungsrate

$$\dot{S} = -Q \cdot \text{sign}(S), \quad (3.20)$$

2. konstante und proportionale Änderungsrate

$$\dot{S} = -Q \cdot \text{sign}(S) - K \cdot S, \quad (3.21)$$

3. potenzielle Änderungsrate

$$\dot{S} = -Q |S|^\alpha \cdot \text{sign}(S) \quad \text{mit } 0 < \alpha < 1. \quad (3.22)$$

4. Konstante Änderungsrate zweiter Ordnung

$$\ddot{S} = -Q_1 \cdot \text{sign}(S) - Q_2 \cdot \text{sign}(\dot{S}) \quad \text{mit } Q_1 > Q_2. \quad (3.23)$$

Für die in dieser Arbeit durchgeführten Untersuchungen gleitzustandsbasierter Trainingsverfahren für neuronale Netzwerke wird ausschließlich der direkte Ansatz der Schaltfunktion mit Anpassung an zeitdiskrete Systeme verwendet. Dabei wird gezeigt, wie die Anwendung dieses Ansatzes zum Erreichen und Beibehalten eines Gleitzustands höherer Ordnung verwendet werden kann. Untersuchungen zu Reaching-Laws erster und zweiter Ordnung für den in dieser Arbeit dargestellten Flugregler sind in [87–90] zu finden.

3.1.3 Ideale und reales Schalten

Ein bei der praktischen Anwendung von Gleitzustandsreglern wesentlicher und zu berücksichtigender Aspekt ist die nur limitierte Schaltfrequenz gegenüber der theoretischen Annahme von infinit schnellem Schalten. Dies ist bedingt durch die Trägheit von Aktuatorik und Sensorik, sowie der Gegenwart von Störungen und Rauschen. Als primäre Auswirkung dieses verlangsamten Schaltverhaltens stellt sich der Gleitzustand nicht länger direkt

auf dem Schaltregime sondern in einer schmalen Nachbarschaft, Grenzschicht genannt, um das Regime herum ein [75]. Dabei ist die Größe dieses Bereiches umgekehrt proportional zur Schaltfrequenz. Um diese zwei Verhalten zu unterscheiden, werden üblicherweise die Begriffe *idealer* und *realer Gleitzustand* verwendet.

Der sich einstellende Effekt, als Reaktion auf das reale Schaltverhalten mit einer limitierten Schaltgeschwindigkeit, wird in der englischsprachigen Literatur als Chattering bezeichnet. Dabei handelt es sich um ein hochfrequentes Hin- und Herschalten des strukturvariablen Reglers. Durch das in der Realität nicht länger unendlich schnelle, sondern in der Frequenz limitierte Schalten wird das Schaltregime stetig überschritten, was ein sofortiges Zurückschalten zur Folge hat, was wiederum zur Überschreitung in entgegengesetzter Richtung führt. Das daraus resultierende Chattering ist nicht nur in dem hochfrequenten Vorzeichenwechsel der Schaltfunktion zu sehen, sondern wird durch das diskontinuierliche Schaltgesetz auf die Steuergröße übertragen und übt somit einen negativen und nicht selten schädlichen Effekt auf das geschlossene Gesamtsystem aus. Die hohe Frequenz der Aktuatorkommandos kann zu unmodellierten Schwingungen des nichtlinear geregelten Systems führen und zusätzlich eine beschleunigte Abnutzung der mechanischen Komponenten bewirken [72, 75].

Ein weiterer wichtiger Aspekt bei der Unterscheidung zwischen realem und idealen Schalten ist die theoretisch nachweisbare Invarianz des idealen strukturvariablen Systems gegenüber Störungen und unbekannten Parametern [72, 84, 85]. Unter der Randbedingung, dass sich jede interne, wie auch externe Störung des Systems \vec{x} aus Gleichung 3.24

$$\dot{\vec{x}} = \vec{f}(\vec{x}) + \Delta\vec{f}(\vec{x}, \vec{p}, t) + [\mathbf{B}(\vec{x}) + \Delta\mathbf{B}(\vec{x}, \vec{p}, t)] \cdot \vec{u} + \vec{f}_s(\vec{x}, \vec{p}, t) \quad (3.24)$$

mit Hilfe der Eingangsmatrix \mathbf{B} abbilden lässt, kann nach [85] gezeigt werden, dass durch die Verwendung der äquivalenten Regelung die Systemdynamik im Gleitzustand vollständig unabhängig von den auftretenden Störtermen ist. Dabei stellt p einen Vektor mit unbekannten Systemparametern, $\Delta\vec{f}$ und $\Delta\mathbf{B}$ interne, sowie \vec{f}_s externe Störungen dar. Diese theoretische Invarianz gilt jedoch nur für den idealen Gleitzustand und somit nicht für reales Schalten oder die Eintrittsphase vor dem Auftreffen auf das Schaltregime. Eine Ausnahme bildet dabei die Verwendung von den oben gezeigten Reaching-Laws, da diese zwar nicht zur direkten Invarianz des Reaching-Modes führen, jedoch nach [85] zumindest den sogenannten Reaching-Space als Indiz für die Robustheit der Eintrittsphase invariant gegenüber Störungen gestalten.

Die Summe der aufgelisteten Unterschiede zwischen idealem und realem Schaltverhalten bildeten die Grundlage für die Forschung an Methoden, das reale und stark durch Chattering beeinflusste Schaltverhalten für die praktische Anwendung zu optimieren. Ein Ansatz zur Chattering-Reduktion ist die Erzeugung eines kontinuierlichen Steuersignals durch Nutzung von Approximationen der Vorzeichenfunktion, wie beispielhaft der Tangens-Hyperbolicus-Funktion. Diese führt jedoch ebenfalls zur Aufweichung des Schaltgesetzes

in der Form, dass unmodellerte schnelle Dynamiken des Systems zu einer unakzeptablen Regelgüte führen können [75]. Ein weiterer und wesentlich intensiver untersuchter Ansatz ist die Gleitzustandsregelung höherer Ordnung. Durch das Verschieben des diskontinuierlichen Umschaltens in höhere Ableitungen der Schaltfunktion ist es möglich, die Auswirkung des hochfrequenten Schaltens und somit des Chatterings von realen Gleitzustandsreglern in der Steuergröße zu reduzieren oder gänzlich zu unterdrücken [76,77].

Für die Anwendung der Gleitzustandsregelung auf das Training neuronaler Netze ist das Chattering weniger gravierend, da keine mechanischen Komponenten beansprucht werden, sondern ausschließlich die Änderung der Verbindungsgewichte beeinflusst wird. Dennoch werden im folgenden Abschnitt die Grundlagen der Gleitzustandsregelung höherer Ordnung dargestellt, da, auf das Netzwerktraining angewendet, dieses Verfahren neben der Reduktion des Chatterings zusätzlich eine beschleunigte Konvergenz und somit eine gesteigerte Leistungsfähigkeit des Netzwerktrainings aufzeigt [89,91,92].

3.2 Gleitzustandsverfahren höherer Ordnung

Ursprünglich als Verfahren zur Vermeidung des schädlichen Chattering-Effekts vorgestellt, erlaubt die Erzeugung von Gleitzuständen höherer Ordnung die Beibehaltung der charakteristischen Robustheitseigenschaften der Gleitzustandsregelung bei gleichzeitiger Steigerung der Regelgüte im Vergleich zu Ansätzen erster Ordnung [77,91,93]. Übertragen auf dynamische Systeme im *Filipov'schen* Sinne, kann die Ordnung der Gleitzustandsregelung als Grad der Glattheit der Systemdynamik in der Nähe des Gleitregimes verstanden werden [76]. Übertragen auf die Unterscheidung zwischen idealem und realem Schalten, gibt die Gleitordnung darüber hinaus die Größe der Grenzschicht um das eigentliche Gleitregime an. Dabei wird diese Grenzschicht mit steigender Gleitordnung immer kleiner.

Bei Verwendung einer Schaltfunktion zur Definition des zu erreichenden Gleitregimes bei $S(x) = 0$, ist die Gleitordnung k die Anzahl der kontinuierlichen Ableitungen (inklusive der Nullableitung) von S , bis in der k -ten Ableitung das erste Mal Diskontinuität auftritt. Entsprechend kann ein Gleitzustand k -ter Ordnung wie folgt definiert werden:

$$\vec{S} = \dot{\vec{S}} = \ddot{\vec{S}} = \dots = \vec{S}^{(k-1)} = 0. \quad (3.25)$$

Betrachtet man somit erneut das beispielhafte System zweiter Ordnung

$$\ddot{x} = a(t) + b(t) \cdot u \quad (3.26)$$

mit dem Schaltgesetz $u = -k \cdot \text{sign}(S)$, wobei $k > 0$ gilt, und der Schaltfunktion

$$S = \dot{x} + \lambda \cdot x \quad (3.27)$$

ist schnell zu erkennen, dass in der nullten Ableitung von S , wie sie in Gleichung 3.27 zu sehen ist, noch keine Abhängigkeit von der diskontinuierlichen Steuergröße u gegeben ist.

Anders verhält es sich nach einmaligem Ableiten der Schaltfunktion. Für die erste zeitliche Ableitung gilt:

$$\dot{S} = \ddot{x} + \lambda \cdot \dot{x} = a(t) + b(t) \cdot u + \lambda \cdot \dot{x}. \quad (3.28)$$

Durch die Formulierung des Schaltgesetzes ist bereits die erste Ableitung der Schaltfunktion diskontinuierlich, solange die Steuergröße die Dynamik vom Betrag her dominiert. Somit ist nur die nullte Ableitung kontinuierlich, so dass das Beispielsystem entlang der Gleitlinie nach der oben gegebenen Definition aus Gleichung 3.25 eine Gleitordnung von $k = 1$ besitzt. Es ist augenscheinlich, dass bei variierender Definition der Schaltfunktion in Gleichung 3.27, so zum Beispiel bei $S = \lambda \cdot x$, zwei Ableitungen der Schaltfunktion nötig sind, bis die diskontinuierliche Steuergröße in der Gleichung auftaucht und es sich somit um einen Gleitzustand zweiter Ordnung handelt. Allerdings kann für diese Funktion keine Erreichbarkeit des Gleitzustands in finiter Zeit garantiert werden.

Da es sich bei der oben gegebenen Definition der Gleitordnung um die Anzahl der aufeinanderfolgenden Ableitung bis zum Auftauchen von Diskontinuitäten in der Schaltfunktion handelt, ist dies in der Regel mit dem Vorhandensein der diskontinuierlichen Steuergröße in der k -ten Ableitung gleichzusetzen. Dieses Kriterium entspricht der Definition des sogenannten relativen Grades r der nichtlinearen Systemtheorie, mit Bezug auf die Schaltfunktion statt einer sonst üblichen Ausgangsgleichung [38]. Mit Hilfe der Definition der Lie-Ableitung in Form von

$$L_a S(x) = \frac{\partial S}{\partial x} \cdot a(x) \quad (3.29)$$

kann für die r -te Ableitung der Schaltfunktion

$$S^{(r)} = L_a^r S(x) + L_b L_a^{r-1} S(x) \cdot u \quad \text{mit} \quad L_b L_a^{r-1} S(x) \neq 0 \quad (3.30)$$

geschrieben werden. Der relative Grad r bezogen auf die Schaltfunktion gibt somit die Anzahl der aufeinanderfolgenden Ableitungen von S an, bis der Ausdruck $L_b L_a^{r-1} S(x) = \frac{\partial}{\partial u} S^{(r)}$ das erste mal ungleich null ist und somit die Steuergröße u in der Gleichung auftaucht. Für die Betrachtung von Gleitzuständen höherer Ordnung lassen sich nach [77] somit zwei Fälle unterscheiden:

- a) relativer Grad $r = 1 \Rightarrow \frac{\partial}{\partial u} \dot{S} \neq 0$,
- b) relativer Grad $r \geq 2 \Rightarrow \frac{\partial}{\partial u} S^{(i)} = 0 \quad \text{für} \quad i = 1, 2, \dots, r-1 \quad \text{mit} \quad \frac{\partial}{\partial u} S^{(r)} \neq 0$.

Daraus folgt, dass sich ein Gleitzustand höherer Ordnung einstellt, wenn der relative Grad zwischen der Schaltfunktion und der diskontinuierlichen Steuergröße größer als eins ist [75, 76]. In diesem Fall entspricht die Gleitordnung dem relativen Grad in Bezug auf die Schaltfunktion. Anschaulich formuliert, wird also das diskontinuierliche Umschalten in die höheren Ableitungen der verwendeten Schaltbedingung verschoben. So kann beispielsweise neben dem Umschalten am Vorzeichen von x oder S eine weitere Schaltbedingung an

den entsprechenden Ableitungen \dot{x} oder \dot{S} aufgenommen werden. Eine mögliche Umsetzung dieses Vorgehens wird im folgenden Abschnitt anhand des sogenannten Twisting-Algorithmus [76,77] als stabiler Ansatz für einen Gleitzustand zweiter Ordnung vorgestellt. Als Schaltbedingung werden dafür sowohl Ansätze mit Schaltfunktion als auch mit direktem Schalten an den Zuständen x und \dot{x} vorgestellt. Darüber hinaus wird auf die Chattering-Reduktion eines Systems mit dem relativen Grad eins durch die künstliche Erhöhung des relativen Grads eingegangen.

3.2.1 Twisting-Algorithmus

Seinen Namen erhält der Twisting-Algorithmus durch die Erzeugung von wirbelartigen Zustandstrajektorien um das Schaltregime im Phasenraum während der Eintrittsphase. Nach einer erst kürzlich durch [94] vorgeschlagenen Kategorisierung in Generationen, stellt der Twisting-Algorithmus den Beginn der zweiten Generation von Gleitzustandsreglern dar. Als Schaltgesetz für den Twisting-Algorithmus mit Umschaltung an den Zuständen x und \dot{x} folgt beispielsweise:

$$u = -k_1 \cdot \text{sign}(x) - k_2 \cdot \text{sign}(\dot{x}) \quad \text{mit} \quad k_1 > k_2 > 0. \quad (3.31)$$

Für die Bestimmung der Gleitordnung entsprechend der Definition aus Gleichung 3.25 gilt es, die Schaltbedingung so oft abzuleiten, bis die diskontinuierliche Steuergröße auf der rechten Seite der Gleichung auftaucht. Im Fall des Twisting-Algorithmus aus Gleichung 3.31 handelt es sich bei der Schaltbedingung um $x = 0$, so dass analog zu dem Vorgehen bei einer Schaltfunktion, x abgeleitet werden muss, um die Gleitordnung zu bestimmen. Angewendet auf die exemplarische Systemdefinition in 3.26 ist klar zu erkennen, dass erst in der zweiten Ableitung \ddot{x} die diskontinuierliche Steuergröße auftaucht, so dass es sich bei Anwendung des Twisting-Algorithmus auf dieses System um einen Gleitzustand zweiter Ordnung mit entsprechendem relativem Grad zwei handelt. Ein auf diese Weise geregeltes System ist robust gegenüber Störungen sowie Unsicherheiten und im Vergleich zu Gleitzustandsverfahren erster Ordnung frei von Chattering.

Twisting-Algorithmus mit relativem Grad 1

Im Falle eines Systems mit auf die Schaltfunktion bezogenem relativem Grad von eins, kann neben einer klassischen Gleitzustandsregelung erster Ordnung ein Gleitzustand zweiter Ordnung zur Chattering-Reduktion umgesetzt werden. Für das einfache System

$$\dot{x} = a(t) + b(t) \cdot u \quad (3.32)$$

ist dies jedoch nur möglich, indem die Steuergröße u selber Ausgang eines dynamischen Systems erster Ordnung wird [77]. Zu diesem Zweck wird das Schaltgesetz so formuliert, dass es nicht direkt die Steuergröße u beeinflusst, sondern deren erste zeitliche Ableitung \dot{u} .

Mit dem Verstärkungsfaktor K folgt für das Schaltgesetz des Twisting-Algorithmus:

$$\dot{u} = \begin{cases} -u & \text{für } |u| > K \\ -k_1 \cdot \text{sign}(x) - k_2 \cdot \text{sign}(\dot{x}) & \text{für } |u| \leq K. \end{cases} \quad (3.33)$$

Auf diese Weise steuert ein diskontinuierliches \dot{u} den Zustand x in den Ruhezustand und hält ihn dort in einem Gleitzustand zweiter Ordnung, wobei die eigentliche Steuergröße u kontinuierlich ist und somit schädliches Chattering verhindert wird.

Twisting-Algorithmus mit Schaltfunktion

Als letzter betrachteter Fall des Twisting-Algorithmus wird dieser anstatt des direkten Umschaltens an Systemzuständen mit einer Schaltfunktion kombiniert. Wird das Schaltgesetz nun allgemein zu

$$\dot{u} = -r_1 \cdot \text{sign}(S) - r_2 \cdot \text{sign}(\dot{S}) \quad \text{mit} \quad S = \dot{x} + \lambda \cdot x \quad (3.34)$$

abgewandelt, entsteht ein Gleitzustand zweiter Ordnung an dem Gleitregime $S = \dot{S} = 0$. Wie im Schaltgesetz aus Gleichung 3.34 zu sehen, wird erneut nicht direkt die Steuergröße, sondern ihre erste Ableitung durch das diskontinuierliche Schalten beeinflusst. Bei dem durch Gleichung 3.34 erzeugten Gleitregime handelt es sich um eine Linie, welche die Schnittmenge der jeweils durch $S = 0$ und $\dot{S} = 0$ aufgespannten Regime darstellt. Eine abstrakte Veranschaulichung der wirbelnden Trajektorie um diese Linie ist nach [76] in Abbildung 3.7 dargestellt. Ein Nachteil dieser Darstellungsart ist die fehlende Information über die diskontinuierliche zweite Ableitung der Schaltfunktion.

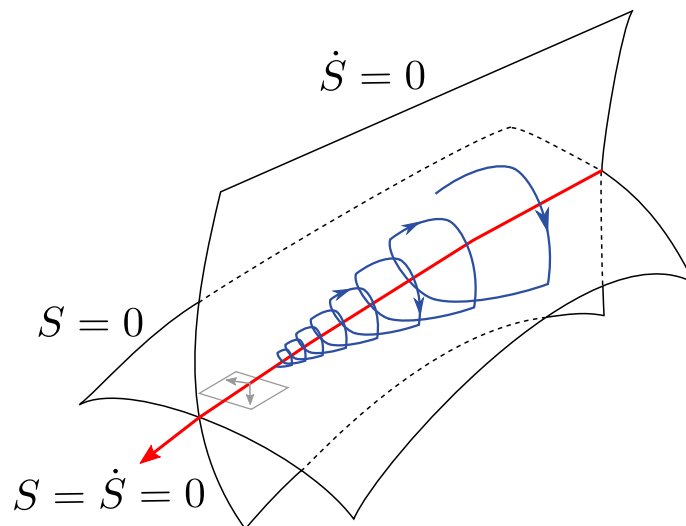


Abbildung 3.7: Darstellung der Zustandstrajektorien des Twisting-Algorithmus im Phasenraum mit Schalten an verschiedenen Schaltregimen nach [76].

So gibt Abbildung 3.7 zwar Aufschluss über den kontinuierlichen Verlaufen der ersten $r - 1$ kontinuierlichen Ableitungen von S , jedoch nicht über das Auftauchen der Diskontinuität

selber. Aus diesem Grund ist in Abbildung 3.8 eine alternative Darstellung für die Trajektorie der Schaltfunktion im dreidimensionalen Phasenraum dargestellt.

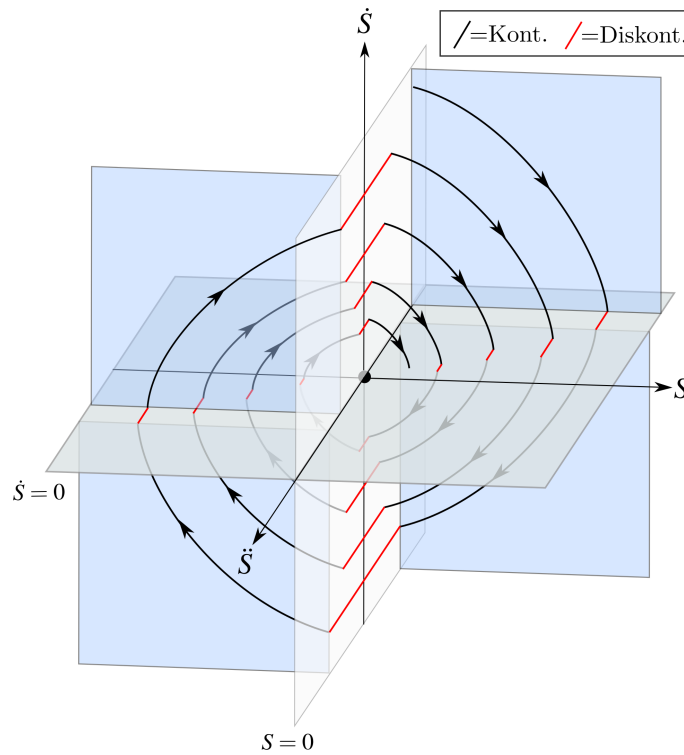


Abbildung 3.8: Abstrakte Darstellung der dreidimensionalen Phasentrajektorie einer Gleitzustandsregelung zweiter Ordnung mit diskontinuierlichem Verhalten in \ddot{S} .

Zu sehen ist das charakteristische wirbelartige Verhalten des Twisting-Algorithmus in der $S - \dot{S}$ -Ebene, wie es auch schon in Abbildung 3.7 dargestellt ist. Darüber hinaus erlaubt es die Darstellung, die diskontinuierlichen Sprünge der zweiten Schaltfunktionsableitung abzubilden. Jedes Mal wenn die Trajektorie die Flächen $S = 0$ oder $\dot{S} = 0$ erreicht, findet ein diskontinuierliches Umschalten von \ddot{S} statt. Durch diese Verschiebung der Diskontinuitäten in die höheren Ableitungen erlaubt es der Twisting-Algorithmus die ersten $r - 1$ Ableitungen kontinuierlich zu halten und somit das schädliche Chattering zu verhindern. Der Übersichtlichkeit halber sind Hilfsebenen für die ersten vier kontinuierlichen Übergänge zwischen $S = 0$ und $\dot{S} = 0$ dargestellt.

Neben dem Twisting-Algorithmus werden in der Literatur verschiedene weitere Algorithmen wie der Sub-optimal-, Drifting- oder Super-Twisting-Algorithmus vorgeschlagen, um eine Gleitzustandsregelung höherer Ordnung zu realisieren [75–77, 91]. Letzterer ist gerade für Systeme geeignet, bei denen es nicht möglich ist, die für das Schalten benötigten höheren Ableitungen der Schaltfunktionen in Echtzeit bereit zu stellen und bildet den Beginn, der dritten Generation von Gleitzustandsreglern [94].

3.2.2 Erreichbarkeits- und Existenzbedingung des Twisting-Algorithmus

Für die Erreichbarkeit und Existenz des durch den Twisting-Algorithmus erzeugten Gleitzustands zweiter Ordnung kann mittels einer Kontraktionsbedingung sowohl $S \cdot \dot{S} < 0$ als auch $S \cdot \ddot{S} < 0$ nachgewiesen werden [75, 76]. Betrachtet wird ein beispielhaftes System der Form

$$\dot{x} = a(t, x) + b(t, x) \cdot u, \quad (3.35)$$

für dessen unbekannte Parameter die folgenden Grenzen gelten:

$$|a(t, x)| \leq C \quad \text{und} \quad 0 \leq K_m \leq b(t, x) \leq K_M. \quad (3.36)$$

Für das Schaltgesetz $u = -r_1 \cdot \text{sign}(S) - r_2 \cdot \text{sign}(\dot{S})$ kann Erreichbarkeit und Existenz des Gleitzustands garantiert werden, solange die Parameter r_1 und r_2 die folgende Bedingung erfüllen:

$$K_m(r_1 + r_2) - C > K_M(r_1 - r_2) + C \quad \text{und} \quad K_m(r_1 - r_2) > C. \quad (3.37)$$

In diesem Fall kann gezeigt werden, dass der Betrag der Schaltfunktionsänderung \dot{S} mit jedem Zeitschritt abnimmt, was zu der gewünschten Kontraktion der Zustandstrajektorie im Phasenraum führt und das System in Richtung des Ursprungs konvergieren lässt [76].

Wie schon in Abschnitt 3.1.2 dargestellt, können auch Reaching-Laws für die Erreichung des Gleitzustands verwendet und darüber hinaus direkt die Dynamik der Eintrittsphase vorgeben werden. Ein erstes Beispiel eines Reaching-Laws zweiter Ordnung mit konstanter Änderungsrate wurde in Gleichung 3.23 gegeben. Eine Möglichkeit für die Formulierung eines Reaching-Laws zweiter Ordnung im Sinne des Twisting-Algorithmus ist die Erweiterung des Ansatzes der potenziellen Änderungsrate aus Gleichung 3.22 [88, 95]. Mit Hinzunahme der Schaltfunktionsänderung \dot{S} und den wählbaren Parametern α und β folgt:

$$\ddot{S} = -Q_1 |S|^\alpha \cdot \text{sign}(S) - Q_2 |\dot{S}|^\beta \cdot \text{sign}(\dot{S}) \quad (3.38)$$

mit

$$0 < \alpha < 1, \quad 0 < \beta < 1, \quad \alpha = \frac{p}{q} \quad \text{und} \quad \beta = \frac{2 \cdot p}{p + q}. \quad (3.39)$$

Dabei müssen die beiden Parameter p und q nach [95] ungerade sein. Darüber hinaus ist es nicht zwingend nötig, dass die Verstärkung Q_1 größer als Q_2 gewählt wird. Für den Fall, dass $Q_1 > Q_2$ gilt, ist ein klassisches, dem Twisting-Algorithmus ähnliches, Twisten der Zustandsgrößen um den Ursprung zu beobachten. Für den Fall $Q_2 > Q_1$ jedoch erreichen die Zustände den stabilen Ruhezustand in finiter Zeit, ohne dabei die charakteristische wirbelartige Bewegung durchzuführen.

Im anschließenden Kapitel werden die oben dargestellten Grundlagen der Gleitzustandsregelung auf das Training neuronaler Netzwerke angewendet. Unter Berücksichtigung der

gezeigten Theorie werden Aussagen getroffen, in welchem Maße das Konzept der strukturvariablen Regelung auf die gezielte Beeinflussung der Fehlerzustände eines neuronalen Netzes angewendet werden kann, um auf diese Weise das Netzwerktraining zu optimieren und bezüglich der regelungstechnischen Anwendung transparenter zu gestalten.

Kapitel 4

Gleitzustandstrainingsverfahren für neuronale Netzwerke

Wie in Kapitel 2 dargestellt, handelt es sich bei neuronalen Netzwerken um lernfähige Elemente, die durch ihre universelle Approximationsfähigkeit in der Lage sind, sich an beliebige nichtlineare Dynamiken anzupassen. Insbesondere vorwärtsgerichtete Netzwerke sind für die Anwendung in adaptiven Regelkreisen geeignet, um sich an variierende Systemeigenschaften bis hin zu Schadensfällen anzupassen und auf diese Weise die Regelgüte zu verbessern. Der Backpropagation-Algorithmus ist auf Grund der nur empirischen Lernratauslegung, der oftmals als intransparent empfundenen Gewichtsänderung, sowie der begrenzten Robustheit gegenüber starken externen Störungen nur eingeschränkt nutzbar, wenn es um die Regelung sicherheitskritischer Systeme geht. Die Robustheit gegenüber internen und externen Störungen, sowie die hohe Leistungsfähigkeit des in Abschnitt 3.1 dargestellten Konzepts der Gleitzustandsregelung sind Motivation, diese mit dem Training neuronaler Netzwerke zu kombinieren. Dabei existieren sowohl Methoden, bei denen neuronale Netzwerke zur Verbesserung von Gleitzustandsreglern eingesetzt [96, 97], als auch Ansätze, bei denen Elemente der Gleitzustandsregelung zur Verbesserung neuronaler Netzwerke verwendet werden [48, 98–101]. Diese Arbeit beschränkt sich ausschließlich auf die Unterstützung des Netzwerktrainings durch ein Gleitzustandsverfahren. Dabei stellen die Robustheit im Gleitzustand, die hohe Konvergenzgeschwindigkeit und die Möglichkeit zur dynamischen Berechnung der Lernrate die wesentlichen Vorteile des Gleitzustandstrainings dar.

Ziel dieses Kapitels ist es, die Anwendung des Gleitzustandsverfahrens auf das Training neuronaler Netzwerke in Hinblick auf die Definition von Schaltgesetzen und Gleitordnungen sowie der gezielten Beeinflussung der Netzwerkzustände darzustellen. Dabei werden die in Abschnitt 3.1 erläuterten Grundlagen dafür verwendet, ein tieferes Verständnis des Lernprozesses eines neuronalen Netzwerks zu vermitteln, indem der direkte Transfer von Erreichbarkeit und Existenz eines Gleitzustands auf die Beeinflussung der Trainingsgröße dargestellt wird. Dabei bilden die Untersuchungen in [48, 66, 102] die Grundlage für die angestellten Analysen.

4.1 Regelung des Netzwerkfehlers

Der Ausgangspunkt des Transfers der Gleitzustandsregelung auf die Anwendung als Trainingsverfahren ist das Verständnis eines neuronalen Netzwerks als dynamisches und zu regelndes System. So wird in [48] von einem Paradigmenwechsel gesprochen, indem das Training eines neuronalen Netzes nicht länger als Optimierungsproblem sondern stattdessen als Regelungsproblem angesehen wird. Schon Ende der 90er Jahre wurde in [99] der Vorteil von, aus der nichtlinearen Regelungstheorie abgeleiteten, Gleitzustandsansätzen für das Training mehrschichtiger Netzwerke dargestellt. Dabei bedarf es zur Anwendung der Gleitzustandsregelung auf das Training neuronaler Netzwerke der Definition der klassischen Größen eines strukturvariablen Regelungssystems, entsprechend der in Kapitel 3.1 dargestellten Theorie. Folglich müssen eine Steuergröße, die Schalfunktion unter Verwendung der zu regelnden Zustandsgröße, die Erreichbarkeits- und Existenzbedingung sowie ein Schaltgesetz definiert werden.

In Abbildung 4.1 ist ein abstraktes Blockschaltbild des als Regelungsproblem verstandenen Netzwerktrainings dargestellt. Als die Systemdynamik beeinflussende Steuergröße wird für das Gleitzustandstraining die Gewichtsänderung $\Delta \mathbf{w}$ genutzt. Mit dem Ziel eines optimalen Trainings ist es naheliegend, die durch die Überführung in den Gleitzustand zu minimierende Größe als Ausgabefehler ε zwischen dem aktuellem Netzwerkausgang y und der Zielgröße y_k zu definieren. Somit ist das Gleitzustandsverfahren erfolgreich, wenn die Zustandsgröße ε im Ruhezustand angekommen ist und dort gehalten wird. Entsprechend wird ε aus Sicht des Trainingsverfahrens auch als Netzwerkzustand bezeichnet.

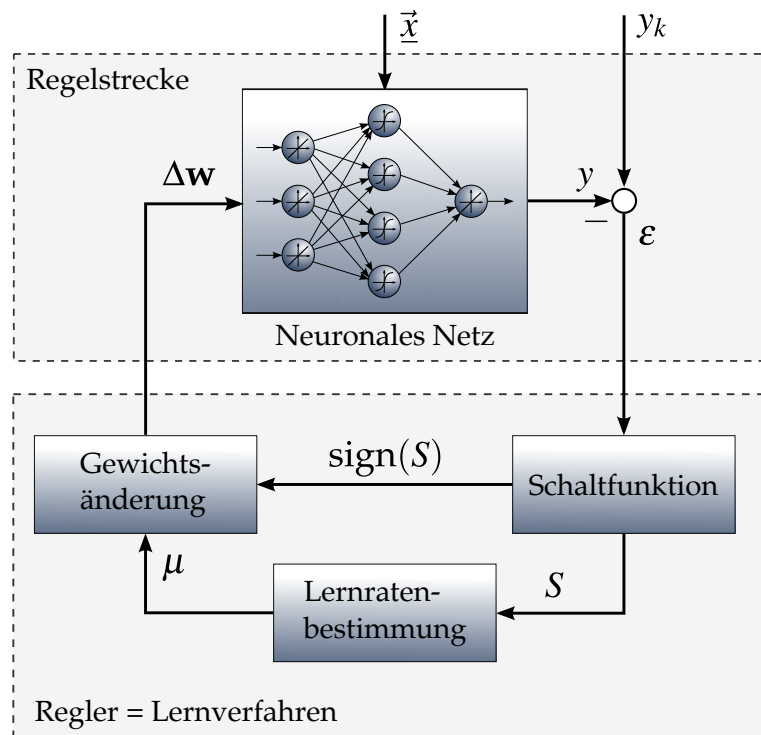


Abbildung 4.1: Das Netzwerktraining als geschlossener Regelkreis.

Wie in der Abbildung zu sehen ist, setzt sich der untere „Regler“-Block aus den drei Elementen Schaltfunktion, Lernratenbestimmung und Gewichtsänderung zusammen. Bei diesen drei Komponenten handelt es sich um die maßgeblichen Freiheitsgrade bei der Auslegung eines Gleitzustandstrainingsverfahrens. Jedes der drei Elemente kann auf verschiedene Weise aufgebaut und mit den jeweils anderen beiden Elementen kombiniert werden. So ist es beispielsweise möglich, verschiedene Auslegungen der Gewichtsänderung mit der selben Schaltfunktion zu kombinieren, aber auch unterschiedliche Schaltfunktionen mit einer gleichbleibenden Lernratenbestimmung zu untersuchen. Insgesamt ergibt sich eine hohe Anzahl von Variationen, von denen in dieser Arbeit nicht alle dargestellt werden können. Aus diesem Grund wird der Fokus auf die dynamische Lernratenbestimmung gelegt, da anhand dieses Elements bereits im Detail die Übertragung der Gleitzustandstheorie auf die Anwendbarkeit als Trainingsverfahren dargestellt werden kann. Untersuchungen zur Variation der eigentlichen Schaltfunktion und ihrer Auswirkung auf das Gleitzustandstraining sind in [87, 90, 92] durchgeführt worden.

4.2 Definition der Schaltfunktion

Aus der Gleitzustandstheorie abgeleitet, wird der Ausgabefehler ε als Zustand für die Verwendung in der Schaltfunktion genutzt. Wie in Abschnitt 3.1 beschrieben, werden die in der Schaltfunktion berücksichtigten Zustände im Gleitzustand bei $S = 0$ entlang der Gleitlinie in den Ruhezustand geführt, was im Falle des Gleitzustandstrainings zur Minimierung des Trainingsfehlers führt. In der Literatur finden sich darüber hinaus Beispiele, bei denen nicht der Ausgabefehler ε , sondern stattdessen der quadratische Fehler E [99, 101], die Gradientenmatrix $\nabla E(\mathbf{w})$ [103] oder die Gewichtsänderung $\Delta \mathbf{w}$ [100] in der Schaltfunktion verwendet wird. Durch die Verwendung des Ausgabefehlers ε kann allerdings die Vorzeicheninformation des Fehlers für das Training genutzt werden, was im Vergleich zur Verwendung des quadratischen Fehlers nach [104] zu einer Erhöhung der Konvergenzgeschwindigkeit führt.

Für die Gleitzustandsregelung eines neuronalen Netzes mit Verwendung der ersten n Ableitungen des skalaren Ausgabefehlers ε kann eine allgemeine Definition der Schaltfunktion in der folgenden Form gegeben werden:

$$S = \sum_{i=0}^n \lambda_{n-i} \cdot \varepsilon^{(i)} . \quad (4.1)$$

Dabei sind $\lambda_0 = 1$ und λ_{n-i} Elemente der entsprechenden Hurwitz-Matrix. Für die in dieser Arbeit verwendete Schaltfunktion erster Ordnung mit $n = 1$, folgt somit:

$$S = \dot{\varepsilon} + \lambda \cdot \varepsilon . \quad (4.2)$$

Diese Definition hat die Überführung des Ausgabefehlers und seiner ersten Ableitung in den Ruhezustand und somit zu Null zur Folge. Ein auf diese Weise definiertes Gleitzu-

standstraining konvergiert entlang der vorgegebenen Gleitlinie unter Ausnutzung der charakteristischen Robustheit einer Gleitzustandsregelung. So kann, anders als bei der Betrachtung des Netzwerks als reines Optimierungsproblem, eine klare Vorhersage über die Gewichtsänderung durch die Einhaltung der Erreichbarkeits- und Existenzbedingungen (vgl. Abschnitt 4.3.2) getroffen werden. Ist der Gleitzustand erst erreicht, sind Gewichtsänderungen nur noch in der Form möglich, dass diese zu einem Ausgabebefehl führen, der entlang einer vorgegebenen Zustandstrajektorie in Richtung des stabilen Ursprungs läuft. Im Gleitzustand angekommen, kann das konvergierende Verhalten des Ausgabebefehlers durch die Lösung der Differentialgleichung aus 4.2 für $S = 0$ gezeigt werden:

$$S = \dot{\varepsilon} + \lambda \cdot \varepsilon = 0 \quad \Rightarrow \quad \varepsilon(t) = \varepsilon(t_0) \cdot e^{-\lambda(t-t_0)} . \quad (4.3)$$

Anhand dieser Lösung lässt sich direkt bestimmen, dass für positive λ der Fehler mit der Zeit zu Null konvergiert. Umgekehrt verhält es sich mit negativen Werten von λ , bei dessen Verwendung der Fehler mit der Zeit gegen unendlich divergiert. Eine Veranschaulichung der zeitlichen Lösungen für sowohl $\lambda > 0$ als auch $\lambda < 0$ sind in Abbildung 4.2 dargestellt. Mit der in Gleichung 4.2 definierten Schaltfunktion werden im Anschluss Gleitzustandslernverfahren erster und zweiter Ordnung hergeleitet und untersucht. Dabei wird jeweils auf die verwendete Erreichbarkeits- und Existenzbedingung, sowie die Vorschrift zur Gewichtsänderung eingegangen. Ein besonderer Fokus wird auf die Berechnung der Lernrate gelegt, da diese für die Stabilität des Trainingsverfahrens von größter Wichtigkeit ist und darüber hinaus die Konvergenzgeschwindigkeit des Verfahrens maßgeblich beeinflusst.

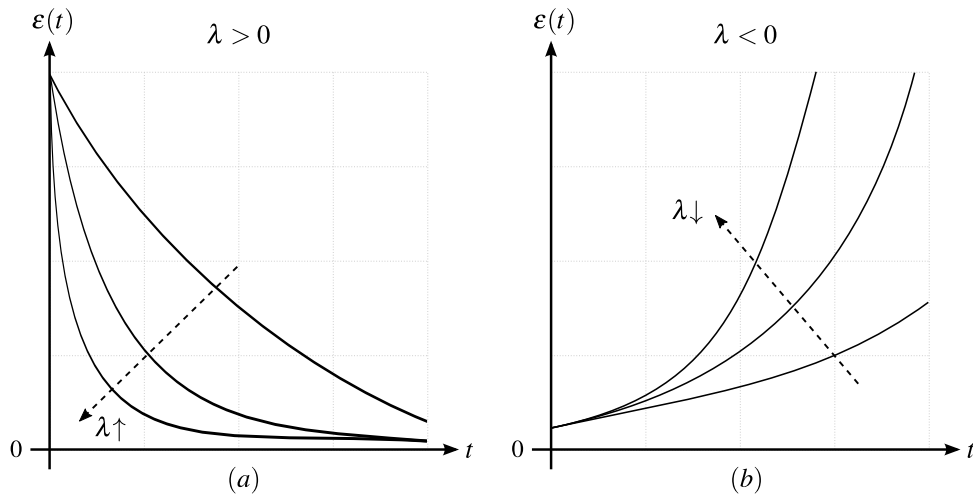


Abbildung 4.2: Lösung der Schaltfunktion für $S = 0$ bei (a) $\lambda > 0$ und (b) $\lambda < 0$.

4.3 Gleitzustandstraining erster Ordnung

Für die in Gleichung 4.2 beschriebene Schaltfunktion muss nun die Existenz und Erreichbarkeit des Gleitzustands nachgewiesen werden. In diesem Abschnitt wird dafür der in Gleichung 3.13 beschriebene Ansatz der direkten Schaltfunktion als Grundlage verwendet. Für

die Anwendung in einem zeitdiskreten System muss die ursprüngliche Bedingung aus [82]

$$\dot{S} \cdot S < 0 \quad (4.4)$$

jedoch angepasst werden. Wie in Abschnitt 3.1.3 dargestellt, führt die Ausführung der Gleitzustandsregelung in realen und zeitdiskreten Systemen nur zum Erreichen eines nicht-idealten Gleitregimes. Da sich dieses quasi-Gleitregime von dem eines kontinuierlichen und idealen Systems unterscheidet, kann es trotz der Verwendung der kontinuierlichen Erreichbarkeits- und Existenzbedingung aus Gleichung 4.4 zu instabilem Verhalten kommen. Das liegt nach [105] darin begründet, dass die durch die Gleitzustandsregelung berechnete Stellgröße im diskontinuierlichen Fall nur in eine Richtung begrenzt ist, so dass es zu theoretisch beliebig großen Steuersignalen kommen kann. Darüber hinaus handelt es sich bei der Bedingung aus Gleichung 4.4 zwar um eine notwendige, jedoch nicht um eine hinreichende Bedingung für die Erreichbarkeit und Existenz eines Gleitzustands. So können Unsicherheiten im System oder zu große Schrittweiten T_s des diskreten Systems zu instabilem Verhalten führen, dass noch immer die Bedingung aus Gleichung 4.4 erfüllt [105, 106]. Zur Veranschaulichung ist in Abbildung 4.3 ein instabiles, sich aufschwingendes System in der Phaseebene dargestellt. Jeder der im oberen-linken oder unteren-rechten Quadranten liegenden Punkte der Systemzustände S_t und \dot{S}_t erfüllen die Bedingung aus Gleichung 4.4. Dennoch ist das Systemverhalten offensichtlich instabil und führt nicht zur Erreichung eines Gleitzustands.

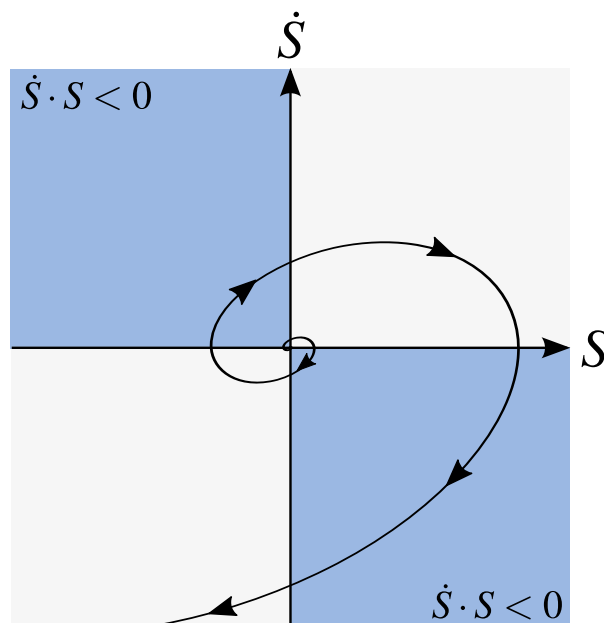


Abbildung 4.3: Phasenportrait eines instabilen diskreten Systems bei Verwendung einer zu großen Schrittweite T_s in Kombination mit einer kontinuierlichen Erreichbarkeitsbedingung.

Aus diesem Grund schlagen [105, 107, 108] eine sowohl hinreichende als auch notwendige Existenz- und Erreichbarkeitsbedingung vor, um auch in einem zeitdiskreten System den

stabilen Gleitzustand garantieren zu können:

$$|S_{t+1}| < |S_t|. \quad (4.5)$$

Dabei wird der Index t dafür verwendet, um den aktuellen Zeitschritt $S(t) = S_t$ und entsprechend $t + 1$, um den nachfolgenden Zeitschritt $S(t + T_s) = S_{t+1}$ zu beschreiben. Unter der Annahme eines geeigneten Stellgesetzes wird so sichergestellt, dass die Systemzustände die Domäne erreichen, in der der Betrag der Schalfunktion mit jedem Zeitschritt abnimmt und sie diesen Bereich anschließend nicht mehr verlassen. Sollte es notwendig werden, kann diese Bedingung abgeschwächt werden, indem auch ein Gleichbleiben des Schalfunktionbetrags für aufeinander folgende Zeitschritte erlaubt wird. Eine ausführliche Darstellung der Anwendung des zeitdiskreten Ansatzes auf das Gleitzustandstraining neuronaler Netzwerke wird weiter unten in Abschnitt 4.3.2 gegeben.

4.3.1 Beschreibung des Stellgesetzes

Bei der Übertragung der Gleitzustandsregelung auf neuronale Lernverfahren, ist es von grundlegender Bedeutung zu untersuchen, auf welche Weise der Vorgang des Schaltens auf die Änderung der Netzwerkzustände transferiert werden kann. Nach der Beschreibung in Kapitel 3 führt das Stellgesetz zu der Steuergröße u , die durch ihre Abhängigkeit von der Schalfunktion die Systemzustände in den Gleitzustand überführt und dort hält. Übertragen auf ein Lernverfahren für neuronale Netzwerke mit dem Netzwerkfehler und dessen Ableitungen als zu regelnde Systemzustände, kann die Gewichtsänderung $\Delta \mathbf{w}$ als Stellgröße aufgefasst werden. Es ist somit erforderlich eine Formulierung zu wählen, die eine Gewichtsänderung in Abhängigkeit der Schalfunktion erlaubt. Dafür ist es zweckmäßig, einen bereits existierenden und etablierten Ansatz der Gewichts Anpassung, wie dem Backpropagation-Algorithmus aus Abschnitt 2.3.1, zu erweitern [48]. Somit ergibt sich eine abgeänderte Gewichts Anpassung für ein Netzwerk mit mehreren Ausgängen und entsprechenden Ausgabebefehlern:

$$\Delta \mathbf{w} = \left(\frac{\partial \vec{y}(\mathbf{w}, \vec{x})}{\partial \mathbf{w}} \right)^T \cdot \mu \cdot \text{diag} \left[\text{sign}(\vec{S}_t) \right] \cdot |\vec{\epsilon}|. \quad (4.6)$$

Umfangreiche Untersuchungen dieses Stellgesetzes und dessen Leistungsfähigkeit sind unter anderem in [48, 66, 90, 102] durchgeführt worden. Eine Alternative dazu stellt die Berechnung der Gewichtsänderung in Abhängigkeit des Gradienten $\frac{\partial (S \cdot \dot{S}_t)}{\partial \mathbf{w}}$ dar [109]. Über das Vorzeichen des berechneten Gradienten hinaus wird das Trainingsverfahren, wie bereits erwähnt, maßgeblich durch die Lernrate μ beeinflusst. Durch die Formulierung der Existenz- und Erreichbarkeitsbedingung können Grenzen für die Lernrate abgeleitet werden, die ein stabiles Training und das Erreichen und Beibehalten eines Gleitzustandes garantieren. Auf diese Weise wird das Stellgesetz, anders als bei der klassischen Gleitzustandsregelung, also nicht nur durch die Schalfunktion, sondern darüber hinaus noch durch die Bestimmung der Lernrate beeinflusst.

4.3.2 Berechnung der Lernrate

In diesem Abschnitt wird die Herleitung einer Berechnungsvorschrift für die Lernrate μ dargestellt. Zu diesem Zweck wird die diskrete Erreichbarkeits- und Existenzbedingung aus Gleichung 4.5 verwendet. Zur Nutzung dieses Ansatzes ist dabei eine Abschätzung der Schaltfunktion im kommenden Zeitschritt S_{t+1} , wie sie in Gleichung 4.5 auftaucht, notwendig. Mit Verwendung des Differenzenquotienten für die erste Ableitung des Netzwerkfehlers $\dot{\vec{e}}_t$

$$\dot{\vec{e}}_t \approx \frac{\vec{e}_t - \vec{e}_{t-1}}{T_s} \quad (4.7)$$

folgt für \vec{S}_t und \vec{S}_{t+1} :

$$\vec{S}_t = \dot{\vec{e}}_t + \lambda \cdot \vec{e}_t = \left(\lambda + \frac{1}{T_s} \right) \cdot \vec{e}_t - \frac{1}{T_s} \cdot \vec{e}_{t-1}, \quad (4.8)$$

$$\vec{S}_{t+1} = \dot{\vec{e}}_{t+1} + \lambda \cdot \vec{e}_{t+1} = \left(\lambda + \frac{1}{T_s} \right) \cdot \vec{e}_{t+1} - \frac{1}{T_s} \cdot \vec{e}_t. \quad (4.9)$$

In Gleichung 4.9 taucht nun der unbekannte Ausdruck \vec{e}_{t+1} für den Netzwerkfehler des folgenden Zeitschritts auf. Dieser wird approximiert, indem der aktuelle Fehler herangezogen und um die Änderung des Fehlers innerhalb eines Zeitschritts erweitert wird [48,66].

$$\vec{e}_{t+1} = \vec{e}_t + \Delta \vec{e}_t \quad (4.10)$$

Die Änderung des Netzwerkfehlers in Gleichung 4.10 kann nun in die unbekannte Änderung des kommandierten Netzausgangs $\Delta \vec{y}_{k,t}$ und die Änderung des tatsächlichen Netzausgangs $\Delta \vec{y}_t$ aufgeteilt werden.

$$\Delta \vec{e}_t = (\vec{y}_{k,t+1} - \vec{y}_{t+1}) - (\vec{y}_{k,t} - \vec{y}_t) = \Delta \vec{y}_{k,t} - \Delta \vec{y}_t. \quad (4.11)$$

Dabei ist es eine Frage der Anwendung der neuronalen Netzwerke, inwieweit eine Aussage über die Änderung des kommandierten Netzausgangs gemacht werden kann. Bei dem Training auf ein vorliegendes Sollsignal kann beispielsweise über eine lineare Extrapolation aus dem letzten Zeitschritt, eine Schätzung der Änderung des kommandierten Netzausgangs berechnet werden. Für die Erweiterung eines Reglers zur bestmöglich *online*-Approximation eines Modellfehlers ist es je nach der Beschaffenheit des vorliegenden Systems schwer, diesen über eine lineare Extrapolation aus der Netzausgabe valide abzubilden. Alternativ kann $\Delta \vec{y}_{k,t}$ zu Null gesetzt werden. Die Änderung des tatsächlichen Netzausgangs $\Delta \vec{y}_t$ kann mit einer Taylorreihenentwicklung erster Ordnung um den Entwicklungspunkt $\mathbf{w}_0 = \mathbf{w}_t$ und $\vec{x}_0 = \vec{x}_t$ angenähert werden [48,101]. Es folgt:

$$\Delta \vec{y}_t = \frac{\partial \vec{y}_t(\mathbf{w}_t, \vec{x}_t)}{\partial \mathbf{w}_t} \cdot \Delta \mathbf{w}_t + \frac{\partial \vec{y}_t(\mathbf{w}_t, \vec{x}_t)}{\partial \vec{x}_t} \cdot \Delta \vec{x}_t. \quad (4.12)$$

Die auf diese Weise entstandenen partiellen Ableitungen $\frac{\partial \vec{y}_t(\mathbf{w}_t, \vec{x}_t)}{\partial \mathbf{w}_t}$ und $\frac{\partial \vec{y}_t(\mathbf{w}_t, \vec{x}_t)}{\partial \vec{x}_t}$ sind aus der Berechnung des allgemeinen Backpropagation-Algorithmus bekannt. Dabei handelt

es sich bei der Ableitung des Netzausgangs \vec{y}_t nach den Netzwerkgewichten \mathbf{w}_t um die Jacobi-Matrix $\mathbf{J}(\mathbf{w})$, vgl. 2.10. Die Änderung der Netzwerkeingänge für den kommenden Zeitschritt \vec{x}_t kann je nach ihrer Beschaffenheit durch geeignete Verfahren angenähert und geschätzt werden. Für kleine Zeitschritte T_s und eine stetige Bewegung durch den Zustandsraum kann diese alternativ auch als vernachlässigbar klein angenommen werden.

Mit den vorgestellten Ausdrücken für die Berechnung des Netzwerkfehlers des folgenden Zeitschritts $\vec{\epsilon}_{t+1}$ ergibt sich nun eine vollständig bekannte Formulierung für die Berechnung der Schalfunktion im kommenden Zeitschritt \vec{S}_{t+1} :

$$\vec{S}_{t+1} = \left(\lambda + \frac{1}{T_s} \right) \vec{\epsilon}_{t+1} - \frac{1}{T_s} \vec{\epsilon}_t = \left(\lambda + \frac{1}{T_s} \right) \left(\vec{\epsilon}_t + \Delta \vec{y}_{k,t} - \frac{\partial \vec{y}_t}{\partial \mathbf{w}_t} \cdot \Delta \mathbf{w}_t - \frac{\partial \vec{y}_t}{\partial \vec{x}_t} \cdot \Delta \vec{x}_t \right) - \frac{1}{T_s} \vec{\epsilon}_t. \quad (4.13)$$

Hier lässt sich nun die Berechnung der Gewichtsänderung aus Gleichung 4.6 einsetzen, um direkt die Abhängigkeit der Schalfunktion im kommenden Zeitschritt von der aktuellen Lernrate ausdrücken zu können.

$$\vec{S}_{t+1} = \left(\lambda + \frac{1}{T_s} \right) \cdot \left(\vec{\epsilon}_t + \Delta y_{k,t} - \frac{\partial \vec{y}_t}{\partial \mathbf{w}_t} \cdot \left[\left(\frac{\partial \vec{y}}{\partial \mathbf{w}} \right)^T \cdot \mu \cdot \text{diag}[\text{sign}(S_t)] \cdot |\vec{\epsilon}| \right] - \frac{\partial \vec{y}_t}{\partial \vec{x}_t} \cdot \Delta \vec{x}_t \right) - \frac{1}{T_s} \vec{\epsilon}_t \quad (4.14)$$

Für eine übersichtlichere Darstellung werden die beiden Parameter a_i und b_i eingeführt. Dabei steht das i für das gewählte Neuron der Ausgabeschicht. Vereinfachend kann für die weiteren Berechnungen in dieser Arbeit von Netzwerken mit nur einem Ausgabeneuron ($i = 1$) ausgegangen werden, da die Netzwerke als Erweiterung des Inversionsregelkreises nur eine Ausgabegröße benötigen.

$$a_i = \left(\lambda + \frac{1}{T_s} \right) \cdot \left(\vec{\epsilon}_{t,i} + \Delta y_{k,t,i} - \left[\frac{\partial y_{t,i}}{\partial \vec{x}_t} \cdot \Delta \vec{x}_t \right] \right) - \frac{1}{T_s} \epsilon_{t,i} \quad (4.15)$$

$$b_i = \left(\lambda + \frac{1}{T_s} \right) \cdot \left(\frac{\partial y_{t,i}}{\partial \mathbf{w}_t} \left(\frac{\partial y_{t,i}}{\partial \mathbf{w}_t} \right)^T \cdot \text{sign}(S_{t,i}) \cdot |\epsilon_i| \right) \quad (4.16)$$

Diese Substitution wird nun in Gleichung 4.14 eingesetzt, sodass sich eine finale Formulierung der Schalfunktion des folgenden Zeitschritts $S_{t+1,i}$ in direkter Abhängigkeit der Lernrate μ der folgenden Form ergibt:

$$S_{t+1,i} = a_i - \mu \cdot b_i. \quad (4.17)$$

Mit Hilfe dieses Ausdrucks ist es möglich, die Lernrate in Abhängigkeit der gewählte Existenz- und Erreichbarkeitsbedingung zu bestimmen, so dass die resultierende Gewichtsänderung zur Reduktion des Netzwerkfehlers durch das Erreichen und Beibehalten des Gleitzustands führt.

Lernratenbestimmung mit diskreter Erreichbarkeitsbedingung

Für die Berechnung einer stabilen und begrenzten Lernrate wird die Bedingung aus Gleichung 4.5 für zeitdiskrete Systeme herangezogen, wobei $|S_{t+1,i}|$ durch die Näherung aus 4.17 ersetzt wird. Somit ergibt sich für die Änderung des Schaltfunktionbetrags in einem Zeitschritt:

$$|a_i - \mu \cdot b_i| < |S_{t,i}|. \quad (4.18)$$

Durch die verwendeten Beträge ist es notwendig eine Fallunterscheidung von Gleichung 4.18 vorzunehmen, um auf diese Weise die Ungleichung nach der Lernrate auflösen zu können. Dabei müssen vier verschiedene Fälle von positiven und negativen Beträgen betrachtet werden. Wie in Tabelle 4.1 zu sehen ist, werden die vier möglichen Fälle nochmal in zwei Gruppen unterteilt. Dabei handelt es sich zum einen um **Fall 1a** und **Fall 1b** bei denen das Vorzeichen der Schaltfunktion im kommenden Zeitschritt unverändert bleibt und zum anderen um die Fälle **Fall 2a** und **Fall 2b** für einen Vorzeichenwechsel von $S_{t,i}$ und $S_{t+1,i}$. Mit anderen Worten: Es muss unterschieden werden, ob die Schaltlinie bei $S_{t,i} = 0$ überschritten wird oder ob sich die Zustände ihr asymptotisch annähern.

Fall	Betrag 1	Betrag 2	Überschreitung der Schaltlinie
Fall 1a:	$S_{t+1,i} > 0$	$S_{t,i} > 0$	Nein
Fall 1b:	$S_{t+1,i} < 0$	$S_{t,i} < 0$	Nein
Fall 2a:	$S_{t+1,i} < 0$	$S_{t,i} > 0$	Ja
Fall 2b:	$S_{t+1,i} > 0$	$S_{t,i} < 0$	Ja

Tabelle 4.1: Fallunterscheidung der zeitdiskreten Existenz- und Erreichbarkeitsbedingung

Fall 1a: $S_{t+1,i} > 0, S_{t,i} > 0$

Wendet man die Forderungen $S_{t,i} > 0$ und $S_{t+1,i} > 0$ auf Gleichung 4.18 an, können die Betragsklammern ohne weitere Veränderungen des Ausdrucks weggelassen werden. Da wie oben dargestellt $S_{t,i}$ und b_i zu jedem Zeitschritt das gleiche Vorzeichen haben, ist es darüber hinaus nicht notwendig an dieser Stelle noch eine Fallunterscheidung an b_i vorzunehmen. Es folgt entsprechend:

$$a_i - \mu \cdot b_i < S_{t,i} \Rightarrow \mu > \frac{a_i - S_{t,i}}{b_i}. \quad (4.19)$$

Es kann somit festgestellt werden, dass für den Fall zweier positiver Beträge in Gleichung 4.18 eine untere Begrenzung der Lernratenberechnung bestimmt werden kann. Die nun benötigte obere Grenze kann durch die Forderung nach einer positiven Schaltfunktion im kommenden Zeitschritt berechnet werden. Aus $S_{t+1,i} > 0$ folgt:

$$a_i - \mu \cdot b_i > 0 \Rightarrow \mu < \frac{a_i}{b_i}. \quad (4.20)$$

Die so gefundene obere Begrenzung für die Berechnung der Lernrate μ ist, anders als die untere Begrenzung, unabhängig von der Schaltfunktion. Folglich nähert sich die untere Grenze bei der Annäherung an die Schaltfläche $S_{t,i} = 0^+$ der oberen Grenze von $\mu < \frac{a_i}{b_i}$ an.

Fall 1b: $S_{t+1,i} < 0, S_{t,i} < 0$

Für den Fall gleichbleibend negativer Vorzeichen der aktuellen und künftigen Schaltfunktion, so dass $S_{t,i} < 0$ und $S_{t+1,i} < 0$ gilt, folgt durch das Wissen um $b_i < 0$ und beidseitiges Ergänzen eines Minuszeichens in Gleichung 4.18:

$$-a_i + \mu \cdot b_i < -S_{t,i} \Rightarrow \mu > \frac{a_i - S_{t,i}}{b_i}. \quad (4.21)$$

Es handelt sich um die gleiche untere Grenze wie in Gleichung 4.19 für **Fall 1a**. Im Anschluss wird durch $S_{t+1,i} < 0$ die dazu gehörige obere Begrenzung berechnet.

$$a_i - \mu \cdot b_i < 0 \Rightarrow \mu < \frac{a_i}{b_i} \quad (4.22)$$

Auch dieser Ausdruck entspricht der Begrenzung aus Gleichung 4.20 des zuerst betrachteten Falls. Es kann also festgehalten werden, dass die Bestimmung einer Lernrate innerhalb des Intervalls $\frac{a_i - S_{t,i}}{b_i} < \mu < \frac{a_i}{b_i}$ nicht zum Überschreiten der Schaltlinie führt, sondern in Abhängigkeit des aktuellen Vorzeichens der Schaltfunktion, diese aus der gleichen Richtung näher an die Schaltlinie heran führt.

Fall 2a: $S_{t+1,i} < 0, S_{t,i} > 0$

Für den Fall einer positiven Schaltfunktion im aktuellen und einer negativen Schaltfunktion im kommenden Zeitschritt bedarf es der Multiplikation mit minus Eins von ausschließlich der rechten Seite in Gleichung 4.18. Es folgt:

$$-a_i + \mu \cdot b_i < S_{t,i} \Rightarrow \mu < \frac{a_i + S_{t,i}}{b_i}. \quad (4.23)$$

Dies führt ganz offensichtlich zu der Berechnung einer oberen Grenze der Lernratenbestimmung. Im Vergleich zu der unteren Grenze aus Gleichung 4.19 ist erneut der Ausdruck $\frac{a_i}{b_i}$ die Basis der Begrenzung, welche sich durch den Wert der Schaltfunktion verschiebt, in diesem Fall jedoch mit umgekehrten Vorzeichen. Für die Berechnung der zweiten benötigten Grenze wird erneut die zweite Ungleichung aufgestellt. Aus $S_{t+1,i} < 0$ ergibt sich:

$$a_i - \mu \cdot b_i < 0 \Rightarrow \mu > \frac{a_i}{b_i}. \quad (4.24)$$

Es handelt sich um die benötigte untere Begrenzung, welche der oberen Grenze für den **Fall 1a** entspricht.

Fall 2b: $S_{t+1,i} > 0, S_{t,i} < 0$

Für den final betrachteten Fall, dass $S_{t,i} < 0$ und $S_{t+1,i} > 0$, sowie durch das Vorzeichen von $S_{t,i}$ ebenfalls vorgegebene $b_i < 0$ gilt, folgt aus Gleichung 4.18:

$$a_i - \mu \cdot b_i < -S_{t,i} \Rightarrow \mu < \frac{a_i + S_{t,i}}{b_i}. \quad (4.25)$$

Es handelt sich somit um die obere Grenze, wie sie auch in **Fall 2a** bestimmt wurde. Erneut wird die Bedingung für das Schaltfunktionsvorzeichen im nächsten Zeitschritt für die Berechnung der zweiten Grenze verwendet:

$$a_i - \mu \cdot b_i > 0 \Rightarrow \mu > \frac{a_i}{b_i}. \quad (4.26)$$

Dabei handelt es sich ebenfalls um die Grenze, die schon in **Fall 2a** gefunden wurde. Es kann folglich festgestellt werden, dass die Wahl einer Lernrate im Intervall $\frac{a_i}{b_i} < \mu < \frac{a_i + S_{t,i}}{b_i}$, unabhängig vom aktuellen Vorzeichen der Schaltfunktion, zu einem Vorzeichenwechsel dieser im kommenden Zeitschritt führt. Dabei nähert sich die Lernrate bei kleiner werdendem Abstand zur Schaltlinie dem Wert $\frac{a_i}{b_i}$ an.

Die durch die Unterscheidung der vier Fälle aufgestellten Grenzen können nun innerhalb eines Vereinigungsintervalls zusammengefasst werden. Dabei wird schnell ersichtlich, dass sich die zwei gefundenen Intervalle am Wert $\frac{a_i}{b_i}$ treffen und der Term $\frac{S_{t,i}}{b_i}$ das Intervall jeweils nach oben und unten aufspannt. Somit kann die Lösungsmenge für die Lernratenbestimmung unter der Bedingung $|a_i - \mu \cdot b_i| < |S_{t,i}|$ wie folgt berechnet werden:

$$\mathcal{L} = \left\{ \mu \in \mathbb{R} \mid \frac{a_i}{b_i} - \frac{S_{t,i}}{b_i} < \mu < \frac{a_i}{b_i} + \frac{S_{t,i}}{b_i} \right\} = \left(\frac{a_i - S_{t,i}}{b_i}, \frac{a_i + S_{t,i}}{b_i} \right). \quad (4.27)$$

In Abbildung 4.4 wird das Prinzip der Vereinigung der unterschiedlichen Fälle grafisch dargestellt. Dabei werden die beiden jeweils nach oben oder unten durch $\frac{a_i}{b_i}$ begrenzten Teilabschnitt in einem großen Lösungsintervall vereinigt. Es ist gut zu erkennen, dass für das neuronale Netzwerk kurz vor dem Gleitzustand und unabhängig von der Richtung aus der die Zustände auf die Schaltfläche $S_{t,i} = 0$ treffen, die Lösung für die Lernrate gegen $\mu = \frac{a_i}{b_i}$ konvergiert.

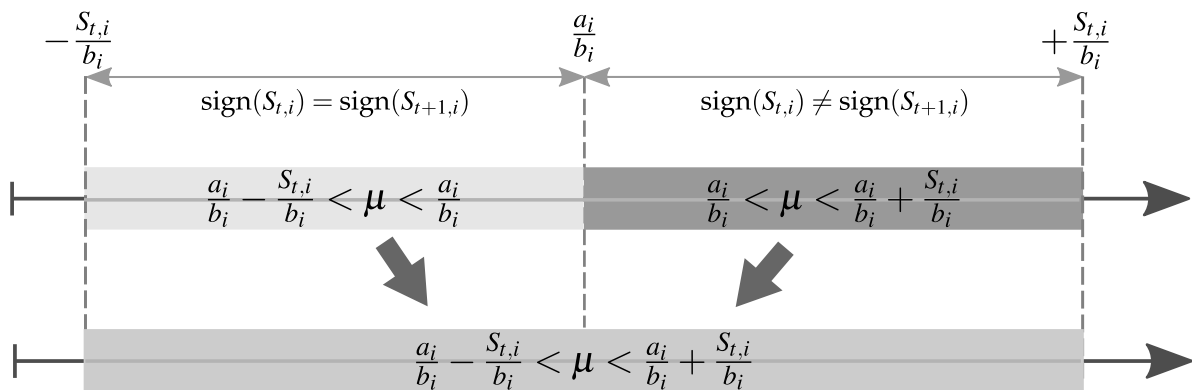


Abbildung 4.4: Lösungsintervall der Lernratenberechnung für die diskrete Erreichbarkeits- und Existenzbedingung $|S_{t+1}| < |S_t|$

Um eine Beschreibung der Lernrate für das Netzwerk im Gleitzustand zu finden, werden die Bedingung für $S_{t,i}$ und $S_{t+1,i}$ in den vier beschriebenen Fällen so gelockert, dass auch Gleichheit anstatt ausschließlich strikter Ungleichheit betrachtet wird [105]. Dies führt zu

der finalen Beschreibung des Lösungsintervalls:

$$\mathcal{L} = \left\{ \mu \in \mathbb{R} \mid \frac{a_i}{b_i} - \frac{S_{t,i}}{b_i} \leq \mu \leq \frac{a_i}{b_i} + \frac{S_{t,i}}{b_i} \right\} = \left[\frac{a_i - S_{t,i}}{b_i}, \frac{a_i + S_{t,i}}{b_i} \right]. \quad (4.28)$$

Mit der in Gleichung 4.28 gegebenen Vorschrift für die stabile Berechnung der Lernrate kann das Netzwerk nun so trainiert werden, dass der Netzwerkfehler in den Gleitzustand überführt wird und in diesem in den stabilen Ursprung gleitet. Auf diese Weise ist es im Vergleich zum klassischen Backpropagation-Algorithmus nicht notwendig, eine empirische Auslegung einer, für das jeweilige Trainingsproblem geeigneten, Lernrate vorzunehmen. Die daraus resultierende Flexibilität gegenüber sich ändernden Prozesseigenschaften bei weiterhin garantierter Stabilität des Trainings, stellt einen der wesentlichen Vorteile des vorgestellten Lernverfahrens dar. Zur Veranschaulichung des Gleitzustandstrainings erster Ordnung mit diskreter Erreichbarkeits- und Existenzbedingung sowie dynamischer Lernratenbestimmung ist in Abbildung 4.5 das erweiterte Blockschaltbild für ein neuronales Netzwerk mit nur einer Ausgangsgröße dargestellt.

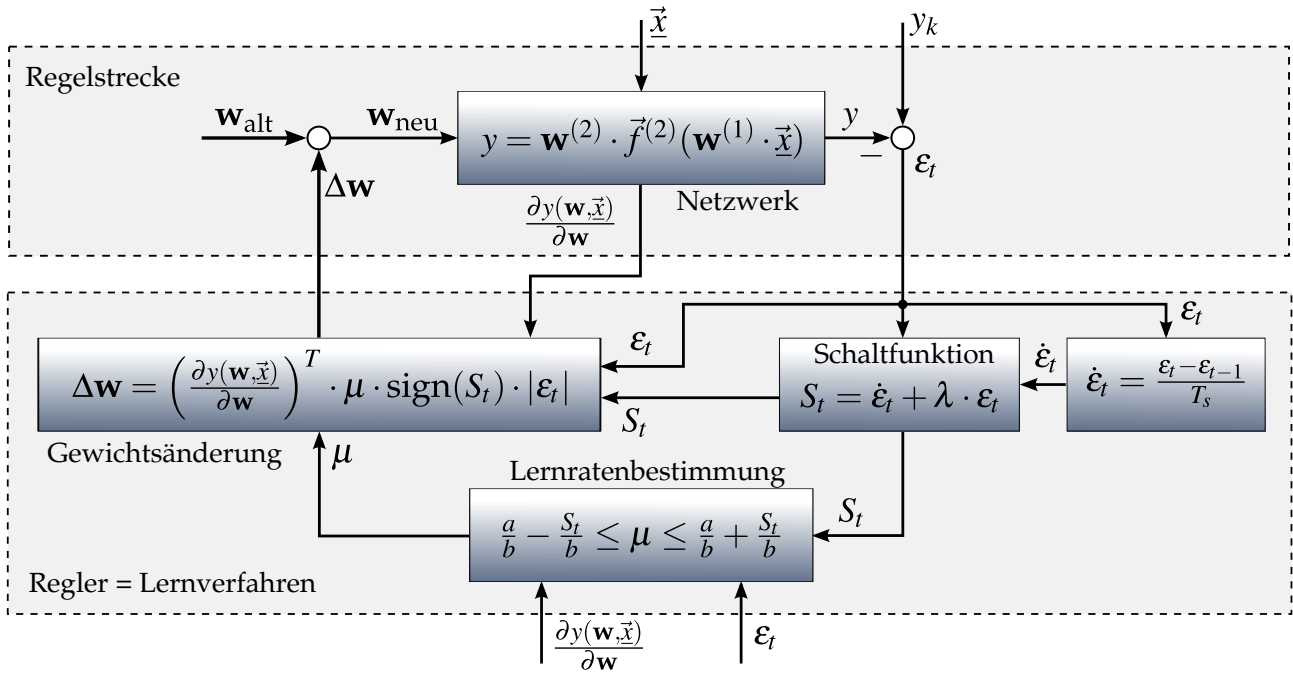


Abbildung 4.5: Blockschaltbild des Gleitzustandstrainings erster Ordnung für ein Netzwerk mit einer Ausgangsgröße

Um das oben beschriebene Netzwerktraining durchführen zu können, muss für jeden Zeitschritt eine Lernrate innerhalb des dynamisch variierenden Intervalls aus Gleichung 4.28 ausgewählt werden. Zu diesem Zweck wird der sogenannte Intervallparameter τ eingeführt. Dieser ist in einem Wertebereich von $0 \leq \tau \leq 1$ definiert und gibt die Position der gewählten Lernrate innerhalb des Intervalls an. Dabei gibt $\tau = 0$ die untere und $\tau = 1$ die obere Grenze an. Zur Verdeutlichung ist in Abbildung 4.6 eine entsprechende Skala entlang des Lernratenintervalls dargestellt.

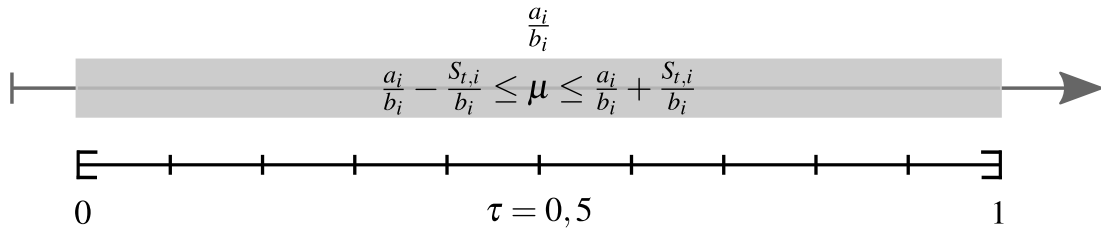


Abbildung 4.6: Darstellung des Intervallparameters τ zur Angabe einer Position im Lernratenintervall

Aus der Herleitung des Intervalls auf Grundlage der in Tabelle 4.1 dargestellten Fallunterscheidung ergibt sich sofort, dass für $\tau < 0,5$ das Vorzeichen der Schaltfunktion vom aktuellen zum kommenden Zeitschritt gleich bleibt. Entsprechend führt die Wahl von $\tau > 0,5$ zum Wechsel des Schaltfunktionsvorzeichens innerhalb eines Zeitschritts und somit zum Überschreiten der Schaltlinie. Für $\tau = 0,5$ wird exakt die Mitte des Intervalls ausgewählt, was der Berechnung einer Lernrate von $\mu = \frac{a_i}{b_i}$ entspricht. Aus der Definition von S_{t+1} in Gleichung 4.17 geht in diesem Fall hervor, dass $\mu = \frac{a_i}{b_i}$ genau den Fall beschreibt, in dem die Schaltfunktion im kommenden Zeitschritt zu null wird. Theoretisch führt die Wahl einer Lernratenberechnung mit $\tau = 0,5$ somit zu einem direkten Erreichen des Gleitzustands im nächsten Zeitschritt.

Mit Hilfe des Intervallparameters τ kann nun eine vereinfachte Berechnungsvorschrift für die Lernrate aus Gleichung 4.28 formuliert werden. Mit den Parametern a_i und b_i sowie dem Wert der aktuellen Schaltfunktion $S_{t,i}$ folgt für die Lernrate:

$$\mu = \frac{a_i}{b_i} + \frac{(2 \cdot \tau - 1) \cdot S_{t,i}}{b_i} \quad \text{mit} \quad 0 \leq \tau \leq 1. \quad (4.29)$$

Mit der in Gleichung 4.29 gegebenen Formulierung der Lernrate lässt sich nun ein Verhältnis zwischen dem aktuellem Wert der Schaltfunktion und dem des folgenden Zeitschritts angeben. Auf diese Weise kann sehr anschaulich der Effekt des Intervallparameters τ auf die Erreichung des Gleitzustands dargestellt werden. Mit $-S_{t+1,i} = -a_i + \mu \cdot b_i$ folgt aus 4.29:

$$\begin{aligned} \mu &= \frac{a_i}{b_i} + \frac{(2 \cdot \tau - 1) \cdot S_{t,i}}{b_i}, \\ -a_i + \mu \cdot b_i &= (2 \cdot \tau - 1) \cdot S_{t,i}, \\ \frac{S_{t+1,i}}{S_{t,i}} &= -(2 \cdot \tau - 1). \end{aligned} \quad (4.30)$$

Bei dem hier Konvergenzquotient genannten Ausdruck $\frac{S_{t+1,i}}{S_{t,i}}$ handelt es sich um einen einfachen linearen Zusammenhang, so dass mit dem durch τ vorgegebenen Definitionsbereich von $[0; 1]$ ein Wertebereich von $[-1; 1]$ abgedeckt wird. Der Konvergenzquotient kann dabei als Verzögerung der Erreichung des Gleitzustands verstanden werden, so dass ein Wert von Null folglich das schnellste Eintreffen auf der Gleitlinie beschreibt. Daraus geht direkt hervor, dass $\tau = 0$ und $\tau = 1$ die Grenzfälle darstellen, bei denen es zu keiner Konvergenz

des Trainings mehr kommen kann. Mit einem Konvergenzquotienten von $\frac{S_{t+1,i}}{S_{t,i}} = 1$ beziehungsweise $\frac{S_{t+1,i}}{S_{t,i}} = -1$ bleibt der Schaltfunktionsbetrag in jedem Zeitschritt gleich, so dass es zu keiner Annäherung an den Gleitzustand kommt. In direkter Konsequenz lässt sich also festhalten, dass obwohl die Wahl eines Intervallparameters von $\tau = 1$ zu der nominal größten Lernrate führt, die Konvergenzgeschwindigkeit der Gleitzustandserreichung durch diesen an der Stabilitätsgrenze gewählten Wert maßgeblich reduziert wird. In Tabelle 4.2 sind die Werte des Konvergenzquotienten für eine Auswahl von Intervallparameter gegeben.

τ	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
$\frac{S_{t+1,i}}{S_{t,i}}$	1	0,8	0,6	0,4	0,2	0	-0,2	-0,4	-0,6	-0,8	-1

Tabelle 4.2: Konvergenzquotient für ausgewählte Intervallparameter

Wichtig zu beachten ist, dass die Lernrate durch die oben beschriebene Berechnung sowohl positive als auch negative Werte annehmen kann. Dies stellt einen deutlichen Kontrast zu der sonst zwingend positiven Schrittweite eines klassischen Gradientenabstiegsverfahrens dar. Allerdings basiert die Berechnung der Lernrate für das hier gezeigte Gleitzustandstraining auf einer Gewichtsänderung $\Delta \mathbf{w}$, in der das Vorzeichen sowohl der Lernrate als auch der Schaltfunktion berücksichtigt wird.

$$\Delta \mathbf{w} = f(\text{sign}(\mu), \text{sign}(S)) \quad (4.31)$$

So gibt die Kombination der Vorzeichen von Lernrate und Schaltfunktion stets die für das Erreichen und Halten des Gleitzustands nötige Richtung vor. Folglich ist die Wahl einer negativen Lernrate nach der oben gezeigten Herleitung zulässig, da die Berechnung direkt der Stabilitätsbedingung entspringt und es somit für den jeweiligen Zeitschritt ein zumindest notwendiges Kriterium für die stabile Konvergenz des Trainingsverfahrens darstellt. Bei der Berechnung einer sowohl unteren als auch oberen Begrenzung des Lernratenintervalls mit negativem Vorzeichen wird eine negative Lernrate sogar zum hinreichenden Kriterium für die Erreichbarkeit und Existenz des Gleitzustands. So würde der aus der allgemeinen Gradientenabstiegstheorie für den Einsatz als Trainingsverfahren abgeleitete Zwang einer positiven Lernrate zur Verletzung dieser Bedingung führen. Im folgenden Abschnitt werden der Einfluss negativer Lernraten sowie die Bedingung für deren Berechnung näher dargestellt.

4.3.3 Untersuchung des Gleitzustandstrainings erster Ordnung

Im folgenden Abschnitt soll das Verhalten der Netzwerkzustände im Gleitzustand, der Einfluss der Lernratenberechnung, sowie die Leistungsfähigkeit des Gleitzustandstrainings im direkten Vergleich zum konventionellen Backpropagation-Algorithmus untersucht und dargestellt werden. Zu diesem Zweck wird ein vorwärtsgerichtetes neuronales Netzwerk

mit 5 Eingabe-, 15 verdeckten und einem Ausgabeneuron betrachtet. Dabei werden die Startgewichte einmalig zufällig bestimmt und anschließend für jede der durchgeführten Simulationen verwendet, um eine direkte Vergleichbarkeit der Ergebnisse zu gewährleisten.

Um die Überführung des Ausgabefehlers ε in den Gleitzustand und damit den Nachweis der Übertragbarkeit der Gleitzustandsregelung auf das Training neuronaler Netzwerke darzustellen, wird zunächst die Approximation eines zweifachen Sollgrößensprungs untersucht. Zu diesem Zweck wird der Sollgrößenverlauf aus Abbildung 4.7 verwendet.

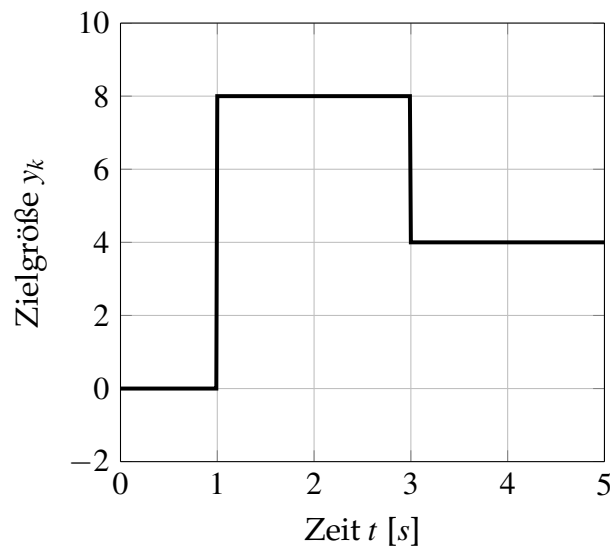


Abbildung 4.7: Zielgröße des Netzwerktrainings mit zweifachem Sprung

Für dieses einfache Trainingsszenario bezieht das Netzwerk ausschließlich Eingabegrößen mit konstanten Zahlenwerten, so dass Sprünge der Netzwerkausgabe ausschließlich auf die durch das Training erzeugten Gewichtsänderungen zurückgeführt werden können. Für den Nachweis der Erreichbarkeit und Existenz eines Gleitzustands der Netzwerkzustände und der Auswirkung unterschiedlicher Intervallparameter τ auf das Netzwerktraining, ist in Abbildung 4.8 das Phasenportrait für ε und $\dot{\varepsilon}$ dargestellt. Dabei ist die Darstellung um die aus der Schaltfunktion resultierende Gleitlinie mit einer Steigung von $\lambda = 50$ ergänzt worden. Der Übersichtlichkeit halber ist nicht der vollständige Wertebereich des Intervallparameters dargestellt, sondern nur Werte von $\tau = [0, 2 : 0, 7]$. Es ist deutlich zu erkennen, wie nach den zwei Sprüngen auf $y_k = 8$, beziehungsweise $y_k = 4$ die Zustandstrajektorien aller Netzwerke in Richtung der Gleitlinie streben und entlang dieser in den stabilen Ursprung bei $\varepsilon = \dot{\varepsilon} = 0$ gleiten. Es kann somit festgehalten werden, dass die auf dem Erreichbarkeits- und Existenzkriterium beruhende Lernratenberechnung in Kombination mit der Vorschrift zur Gewichtsänderung ein neuronales Netzwerk in einen stabilen Gleitzustand überführt. Dabei ist in Abbildung 4.8 anschaulich der Einfluss des Intervallparameters τ auf die Annäherung an die Gleitlinie zu erkennen.

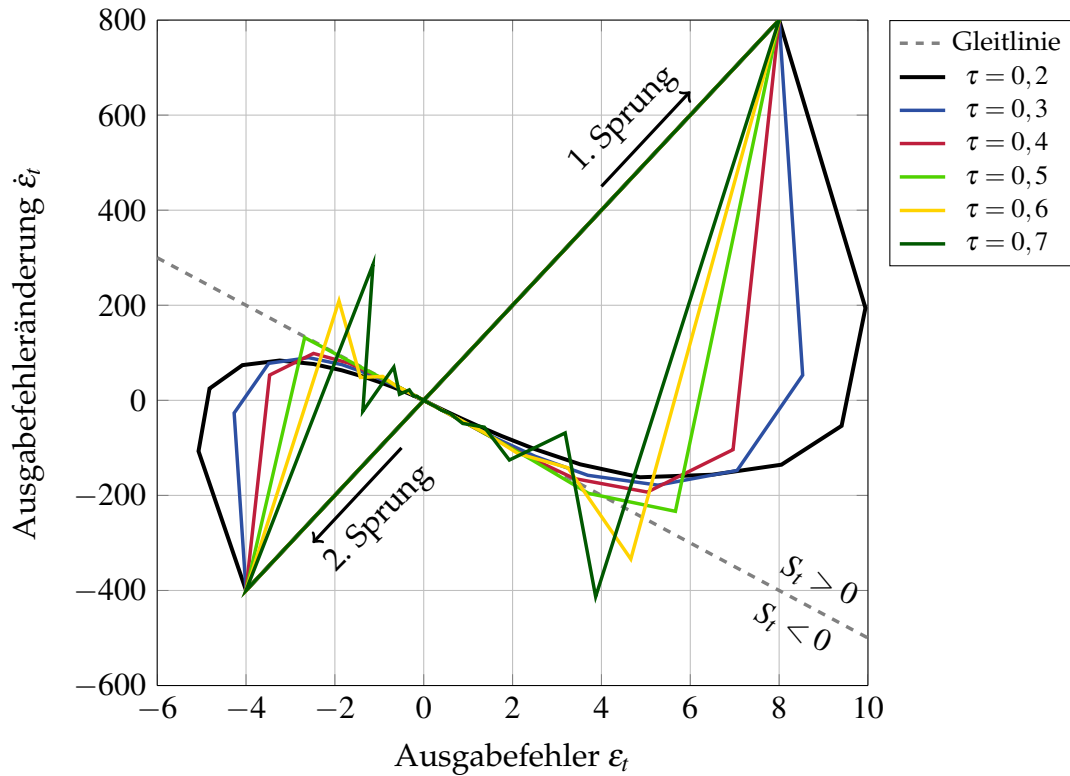


Abbildung 4.8: Phasenportrait eines mit dem Gleitzustandsverfahren erster Ordnung trainierten neuronalen Netzwerks mit zwei Sprüngen der Zielgröße

Wie durch den Konvergenzquotienten $\frac{S_{t+1,i}}{S_{t,i}}$ aus der Theorie abgeleitet, führen Intervallparameter von $\tau < 0,5$ zu einer langsameren Annäherung an die Gleitlinie, ohne diese zunächst zu überschreiten. So bleiben die Netzwerkzustände in der Eintrittsphase in der jeweiligen, durch die Gleitlinie in $S_t > 0$ und $S_t < 0$ aufgeteilten, Hälfte des Phasenraums. Entsprechend gegenteilig verhält es sich für $\tau > 0,5$. Wie durch die Lernratenbestimmung festgelegt, wechselt der Schaltfunktionswert S_t sein Vorzeichen in jedem Zeitschritt und überschreitet während des Trainingsprozesses folglich dauerhaft die Gleitlinie bei $S_t = 0$. Dabei ist gut zu erkennen, dass das Überschießen der Gleitlinie umso stärker ist, desto größer τ gewählt wird. Wie in Tabelle 4.2 abzulesen ist, führt die Wahl von $\tau = 0,5$ zu der schnellsten Annäherung an die Gleitlinie. Wo für den ersten Sprung noch ein minimales Überschießen zu sehen ist, werden im Fall des zweiten Sprungs die Netzwerkzustände für $\tau = 0,5$ direkt nach einem Zeitschritt zurück auf die Gleitlinie geführt. Damit entspricht dieses Verhalten exakt der durch die Lernratenbestimmung vorgegebenen Theorie. Für die detaillierte Betrachtung der Konvergenzgeschwindigkeit sowohl des Ausgabefehlers ϵ als auch der Schaltfunktion S_t sind in Abbildung 4.9 die entsprechenden Zeitverläufe für den ersten Zielgrößensprung dargestellt. Aus Gründen der Übersichtlichkeit werden dabei nur drei exemplarische Werte des Intervallparameters abgebildet.

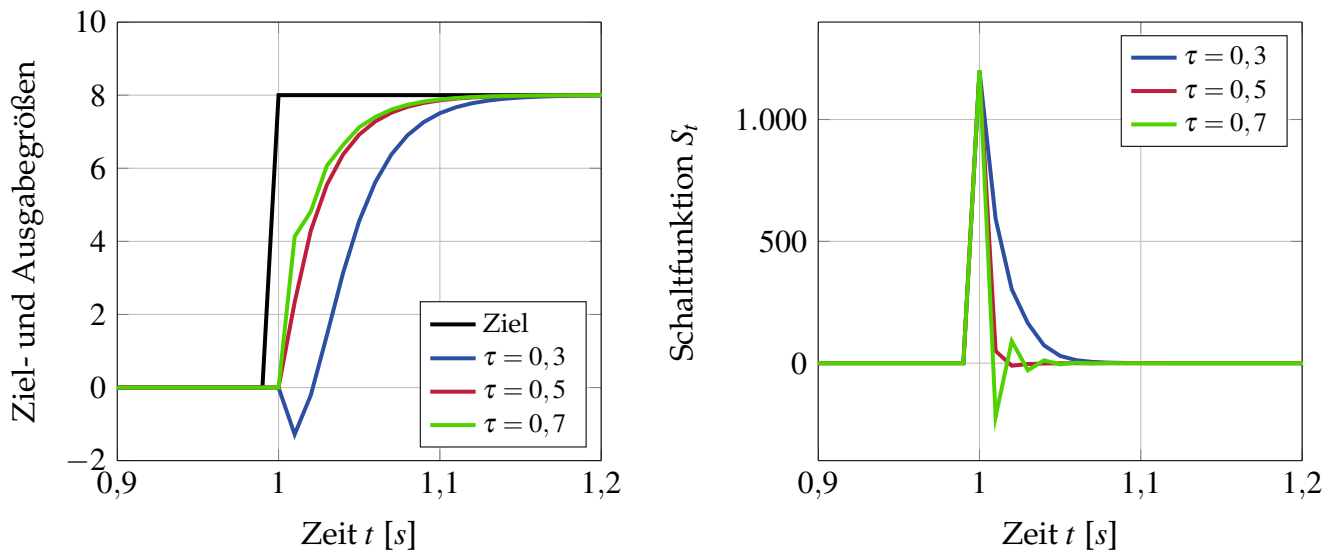


Abbildung 4.9: Netzausgaben und Schaltfunktionswerte für verschiedene Intervallparameter bei einem Sprung der Zielgröße

Wie auch in dem Phasenportrait zu sehen, gelingt es allen Netzwerken den Sprung der Zielgröße auszugleichen und auf diese Weise den Ausgabefehler schnell und stabil abzubauen. Dabei fallen jedoch verschiedene Besonderheiten bei den unterschiedlichen Intervallparametern auf. Besonders markant ist die zunächst in die entgegengesetzte Richtung zunehmende Netzwerkausgabe für $\tau = 0,3$. Dieser unerwünschte Fall tritt auf, wenn die Schaltfunktion zwar durch die Kombination aus Fehleränderung $\dot{\epsilon}$ und Fehler ϵ für jeden Zeitschritt betragsmäßig kleiner wird, dabei jedoch der Anteil von $\dot{\epsilon}$ so groß ist, dass ein Anstieg des Fehlers möglich bleibt. Besagtes Verhalten kann durch den Vergleich mit dem dargestellten Zeitverlauf der Schaltfunktion in Abbildung 4.9 bestätigt werden. Die Schaltfunktion für $\tau = 0,3$ nimmt trotz zunächst ansteigendem Fehler wie gefordert in jedem Zeitschritt ab.

Eine einfache Methode zur Vermeidung eines zunächst steigenden Fehlers ist die Wahl eines größeren λ in der Schaltfunktion. Auf diese Weise führt jede Änderung des Fehlers zu einer größeren Gesamtbeeinflussung der Schaltfunktion, so dass sich der Betrag von S und des Ausgabefehlers immer ähnlicher werden, bis letztendlich die Bedingung einer abnehmenden Schaltfunktion mit der eines sinkenden Fehlers gleichzusetzen ist. Auf eine zweite Methode, die auch bei kleineren λ wirksam ist, wird weiter unten im Verlauf der Vorzeichendiskussion der Lernrate eingegangen.

Ebenfalls in Abbildung 4.9 zu erkennen ist das Überschießen von $S_t = 0$ für $\tau = 0,7$. Trotz der Überschreitung der Gleitlinie führt die Wahl von $\tau = 0,7$ für das hier gezeigte Trainingsszenario zu einer leicht beschleunigten Konvergenz der Netzwerkausgabe im Vergleich zu den beiden anderen Netzwerken. Dies ist auf die Berechnung von höheren Lernraten durch den größeren Intervallparameter τ zurückzuführen. Über das unterschiedliche Verhalten direkt

nach den Sprüngen hinaus ist in Abbildung 4.9 gut zu erkennen, dass die Fehlerverläufe nach der Annäherung an $S_t = 0$ in Hinblick auf die Dynamik nahezu identisch sind. Dies ist durch das Zusammenrücken der oberen und unteren Grenzen des Lernratenintervalls begründet, so dass sich die Lernraten trotz unterschiedlicher Intervallparameter immer mehr ähneln. Es kann somit gezeigt werden, dass die Dynamik des Fehlerausgleichs nach der Annäherung an den Gleitzustand letztlich nur noch von der Definition der Schaltfunktion und somit von der Wahl von λ abhängt.

Zur Veranschaulichung der Annäherung der mit verschiedenen τ berechneten Lernraten, sind diese in Abbildung 4.10 für den ersten in Abbildung 4.9 gezeigten Sprung dargestellt. Ebenfalls abgebildet ist der zeitliche Verlauf des Schaltfunktionsvorzeichens $\text{sign}(S_t)$ der drei exemplarisch betrachteten Netzwerke.

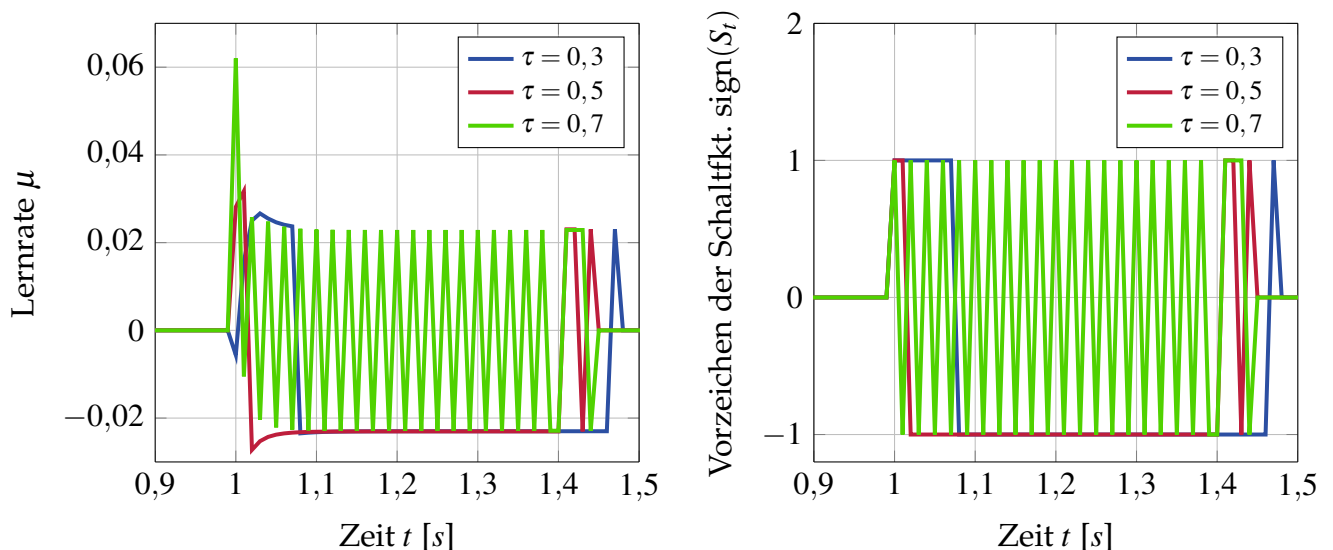


Abbildung 4.10: Lernraten und Schaltfunktionsvorzeichen für verschiedene Intervallparameter bei einem Sprung der Zielgröße

Es ist zu erkennen, dass die Lernraten sich innerhalb der ersten 0,08 Sekunden nach dem Sprung im Betrag und Vorzeichen unterscheiden. Dabei werden für das mit $\tau = 0,7$ trainierte Netzwerk die größte und entsprechend für $\tau = 0,3$ die niedrigste Lernrate bestimmt. Im Anschluss nähern sich die Grenzen soweit aneinander an, dass für alle drei Netzwerke eine Lernrate von $|\mu| = -0,023$ berechnet wird. Auffällig dabei ist das *chattering* des Lernratenvorzeichens für $\tau = 0,7$. Dass es sich dabei um *chattering*, wie es aus der Gleitzustandstheorie bekannt ist, handelt, ist am Verlauf des Schaltfunktionsvorzeichens zu erkennen. Das konstante Überschreiten von $S_t = 0$ und dem daraus resultierende Vorzeichenwechsel führt zu einem ebenfalls konstant alternierendem Vorzeichen der Lernratenberechnung. Des Weiteren ist zu erkennen, dass die Vorzeichen der berechneten Lernraten mit den Vorzeichen der Schaltfunktionen korrelieren. Die einzige Ausnahme bildet der erste Zeitschritt nach dem Sprung für das Netzwerk mit $\tau = 0,3$. In diesem Zeitschritt wird eine negative Lernrate

trotz positivem Betrag der entsprechenden Schaltfunktion berechnet. Als Reaktion darauf ist der Fehleranstieg in Abbildung 4.9 zu sehen.

Auf dieser Grundlage lässt sich, analog zum Vorgehen bei der Herleitung der Lernratenberechnung, ein Kriterium für das Vorzeichen der Lernrate bestimmen. Neben $|S_{t,i}| > |S_{t+1,i}|$ soll so sichergestellt werden, dass der Fehler in jedem Zeitschritt geringer wird. Wie oben erwähnt, ist dies generell durch die Wahl großer λ zu erreichen, da sich auf diese Weise die Beträge von Schaltfunktion und Ausgabefehler annähern. Darüber hinaus lässt sich eine Bedingung für das Vorzeichen der Lernrate aufstellen, die auch für kleine λ gilt. Für die Abnahme des Fehlers in jedem Zeitschritt muss gelten:

$$|\varepsilon_{t,i}| > |\varepsilon_{t+1,i}| . \quad (4.32)$$

Dabei wird an dieser Stelle keine vollständige Fallunterscheidung für die zusätzliche Berechnung einer Lernrate in Abhängigkeit des Fehlers durchgeführt, sondern lediglich eine Vorzeichenkonvention getroffen, die neben der Erreichung des Gleitzustands das Anwachsen des Fehlers bei großer Distanz zum Zielwert, also ohne Überschreitung von $\varepsilon_t = 0$ innerhalb eines Zeitschritts, verhindert. In Tabelle 4.3 sind die zwei betrachteten Fälle eines abnehmenden Fehlers ohne Veränderung des Vorzeichens aufgeführt.

Fall	Betrag 1	Betrag 2	Überschreitung von $\varepsilon = 0$
Fall 1:	$\varepsilon_{t,i} > 0$	$\varepsilon_{t+1,i} > 0$	Nein
Fall 2:	$\varepsilon_{t,i} < 0$	$\varepsilon_{t+1,i} < 0$	Nein

Tabelle 4.3: Fallunterscheidung des Ausgabefehlers

Fall 1: $\varepsilon_{t,i} > 0, \varepsilon_{t+1,i} > 0$

Für den Fall eines sowohl positiven Ausgabefehlers im aktuellen, wie auch im folgenden Zeitschritt folgt mit Gleichung 4.10:

$$\varepsilon_{t,i} > \varepsilon_{t,i} + \Delta\varepsilon_{t,i} . \quad (4.33)$$

Zusammen mit den Gleichungen 4.11 und 4.12, sowie den Näherungen $\Delta y_{k,t,i} = 0$ und $\Delta \vec{x}_t = 0$ kann Gleichung 4.33 umgeschrieben werden zu:

$$\varepsilon_{t,i} > \varepsilon_{t,i} - \frac{\partial \vec{y}_t(\mathbf{w}_t, \vec{x}_t)}{\partial \mathbf{w}_t} \cdot \Delta \mathbf{w}_t . \quad (4.34)$$

Subtrahiert man nun $\varepsilon_{t,i}$ von beiden Seiten, setzt die Vorschrift zur Gewichtsänderung $\Delta \mathbf{w}_t$ aus Gleichung 4.6 ein und löst die Ungleichung unter dem Wissen, dass $\varepsilon_{t,i} > 0$ gilt, nach der Lernrate auf, folgt:

$$0 < \mu \cdot \text{sign}(S_{t,i}) . \quad (4.35)$$

Für den Fall, dass der Ausgabefehler positiv ist und mit jedem Zeitschritt kleiner werden soll, ist es also zwangsläufig notwendig, dass die Lernrate in jedem Zeitschritt das Vorzeichen der Schaltfunktion annimmt.

Fall 2: $\varepsilon_{t,i} < 0, \varepsilon_{t+1,i} < 0$

Für den zweiten Fall mit sowohl negativem Ausgabefehler im aktuellen wie auch im kommenden Zeitschritt folgt nach Auflösen der Beträge:

$$\varepsilon_{t,i} < \varepsilon_{t,i} + \Delta \varepsilon_{t,i} . \quad (4.36)$$

Damit handelt es sich um die Bedingung aus Gleichung 4.33 mit gedrehtem Ungleichheitszeichen. Somit ergibt sich mit dem Wissen um Gleichung 4.35 direkt:

$$0 > \mu \cdot \text{sign}(S_{t,i}) . \quad (4.37)$$

Dies bedeutet, dass bei negativen Fehlern die Lernrate das entgegengesetzte Vorzeichen der Schaltfunktion besitzen muss, damit der Fehler mit jedem Zeitschritt abgebaut wird. Zusammengefasst ergibt sich die folgende Vorzeichenkonvention der Lernrate für die Bedingung eines in jedem Zeitschritt kleiner werdenden Fehlers:

$$\text{sign}(\mu) = \text{sign}(S_{t,i}) \cdot \text{sign}(\varepsilon_{t,i}) = \text{sign}(S_{t,i} \cdot \varepsilon_{t,i}) . \quad (4.38)$$

Unter der Berücksichtigung der Definition des Substitutionsparameters a_i aus Gleichung 4.15 und den oben getroffenen Annahmen $\Delta y_{k,t,i} = 0$ und $\Delta \vec{x}_t = 0$ kann a_i umgeschrieben werden zu:

$$a_i = \left(\lambda + \frac{1}{T_s} \right) \cdot \left(\vec{\varepsilon}_{t,i} + \Delta \vec{y}_{k,t,i} - \left[\frac{\partial \vec{y}_t}{\partial \vec{x}_t} \cdot \Delta \vec{x}_t \right]_i \right) - \frac{1}{T_s} \vec{\varepsilon}_{t,i} = \lambda \cdot \varepsilon_{t,i} . \quad (4.39)$$

Aus Gleichung 4.39 und der Definition der Lernrate mit $\tau = 0,5$ von $\mu = \frac{a_i(\text{sign}(\varepsilon_{t,i}))}{b_i(\text{sign}(S_{t,i}))}$ geht direkt hervor, dass eine derart bestimmte Lernrate die Bedingung aus Gleichung 4.38 zu jeder Zeit erfüllt. So führt die Wahl des Intervallparameters von $\tau = 0,5$ zu monoton abnehmenden Fehlern bei gleichzeitiger Konvergenz der Netzwerkzustände auf die Gleitlinie mit der Einschränkung, dass dies nur für den Fehlerabbau ohne Vorzeichenwechsel, also für große Abstände zur Zielgröße gilt. Die Berechnungsvorschrift der stabilen Lernrate aus Gleichung 4.29 lässt sich somit in der Art ergänzen, dass im Falle der Nichterfüllung von Gleichung 4.38, die Lernrate nicht beim vorgegebenen Intervallparameter sondern in der Intervallmitte bei $\mu = \frac{a_i}{b_i}$ berechnet wird. Zu diesem Zweck wird der Faktor σ^* eingeführt. Dieser kann ausschließlich die Werte 0 und 1 annehmen und legt damit fest, wann die Lernratenberechnung auf $\mu = \frac{a_i}{b_i}$ korrigiert wird. Für die Berechnung der Lernrate mit σ^* -Korrektur in Abhängigkeit der zunächst unkorrigierten Lernrate μ^* folgt somit:

$$\mu = \frac{a_i}{b_i} + \frac{(2 \cdot \tau - 1) \cdot S_{t,i}}{b_i} \cdot \sigma^* \quad \text{mit} \quad \sigma^* = \begin{cases} 1 & \text{für } \text{sign}(\mu^*) = \text{sign}(S_{t,i} \cdot \varepsilon_{t,i}) \\ 0 & \text{für } \text{sign}(\mu^*) \neq \text{sign}(S_{t,i} \cdot \varepsilon_{t,i}) \end{cases} . \quad (4.40)$$

Die angegebene Fallunterscheidung für die Bestimmung von σ^* lässt sich alternativ durch die folgende Gleichung vereinfachen, so dass die Vorschrift zur Bestimmung der Lernrate direkt durch den Ausdruck in Gleichung 4.41 ergänzt werden kann.

$$\sigma^* = |\text{sign}[\text{sign}(\mu^*) + \text{sign}(S_{t,i} \cdot \varepsilon_{t,i})]| \quad (4.41)$$

Mit Hilfe der vorgestellten Korrektur ergeben sich die in Abbildung 4.11 dargestellten Zeitverläufe für das Trainingsszenario, wie es in Abbildung 4.7 vorgestellt wurde.

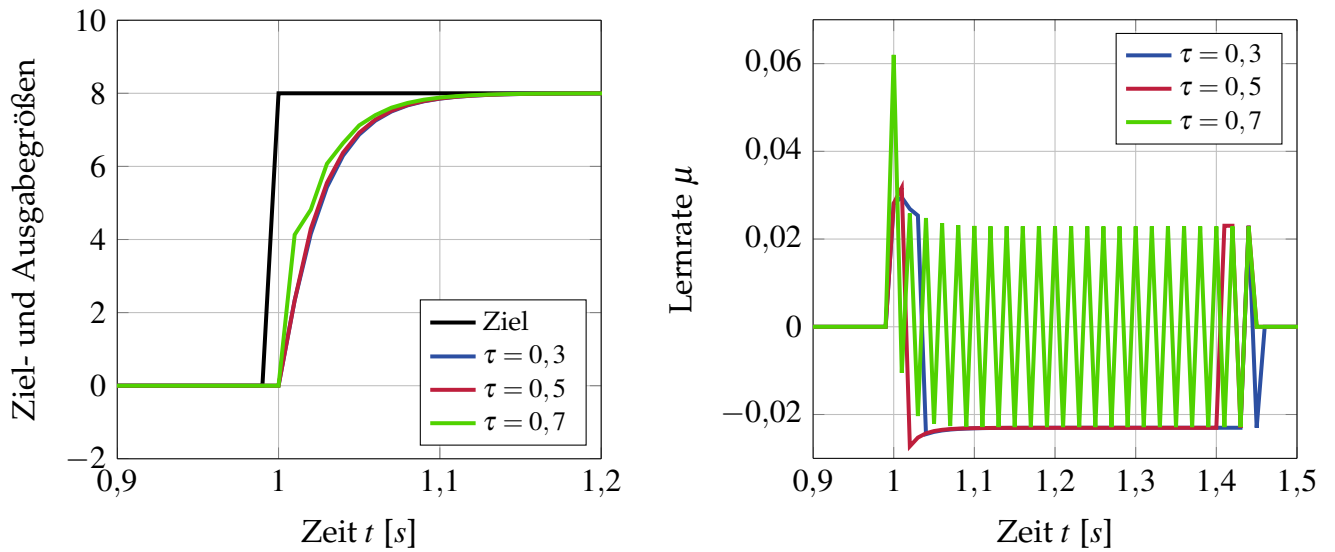


Abbildung 4.11: Netzausgaben und σ^* -korrigierte Lernraten für verschiedene Intervallparameter bei einem Sprung der Zielgröße

Es ist deutlich zu erkennen, dass durch die σ^* -Korrektur der zunächst ansteigende Fehler nach dem Sprung für das Training mit $\tau = 0,3$ verhindert werden kann. Im Zeitverlauf der Lernrate ist im Vergleich zu Abbildung 4.10 nun eine initiale positive Lernrate zu sehen. Aufgrund der Berechnung durch $\mu = \frac{a_i}{b_i}$ hat diese dabei im Zeitschritt der wirksamen σ^* -Korrektur den gleichen Wert wie die Lernrate für das Training mit $\tau = 0,5$.

Für die weitere Untersuchung des Gleitzustandtrainings erster Ordnung und dessen Abhängigkeit von τ und λ , sowie dem Vergleich mit dem klassischen Backpropagation-Algorithmus wird ein komplexeres Trainingsszenario verwendet. Ziel dabei ist es, eine Abstimmung zwischen der theoretisch schnellsten Konvergenz bei $\tau = 0,5$ und größeren Lernraten bei $\tau > 0,5$ zu finden. Das zu erlernende 20 sekundige Trainingssignal setzt sich aus der Kombination einer Sinusschwingung mit überlagerten Sprung-, Rampen- und Impuls-signalen zusammen. Der resultierende nichtlineare Zielgrößenverlauf ist in Abbildung 4.12 dargestellt.

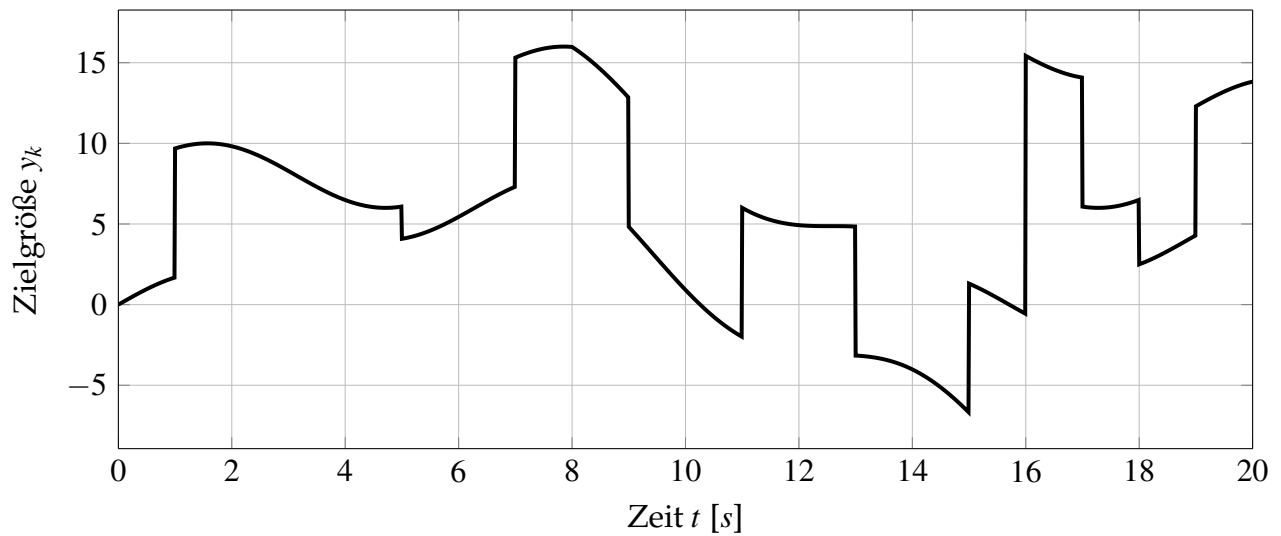


Abbildung 4.12: Komplexer Zielgrößenverlauf des Netzwerktrainings

Als Eingangsgrößen des Netzwerks werden für das zweite Trainingsszenario ebenfalls sinusförmige Schwingungen unterschiedlicher Frequenz, Phase und Amplitude sowie eine Rampe mit variierenden Steigungen verwendet. Auf diese Weise ist im Vergleich zum ersten Trainingsfall nicht ausschließlich die Zielgröße für die Änderung der Netzwerkausgabe verantwortlich. Durch die in keinem direkten Zusammenhang zur vorgegebenen Trainingsgröße stehenden Eingabewerte des Netzwerkes, ist eine dauerhafte Anpassung der Netzwerkgewichte notwendig, um den Ausgabefehler gering und in der Nähe der Gleitlinie zu halten. Der zeitliche Verlauf der fünf Eingangssignale ist in Anhang A in Abbildung A.1 dargestellt. Die Ausgabe y_k dreier mit unterschiedlichen Intervallparametern trainierter Netzwerke mit σ^* -Korrektur für das gezeigte komplexere Trainingsszenario sind in Abbildung 4.13 dargestellt. Dabei wurde erneut $\lambda = 50$ als Steigung der Schaltlinie verwendet.

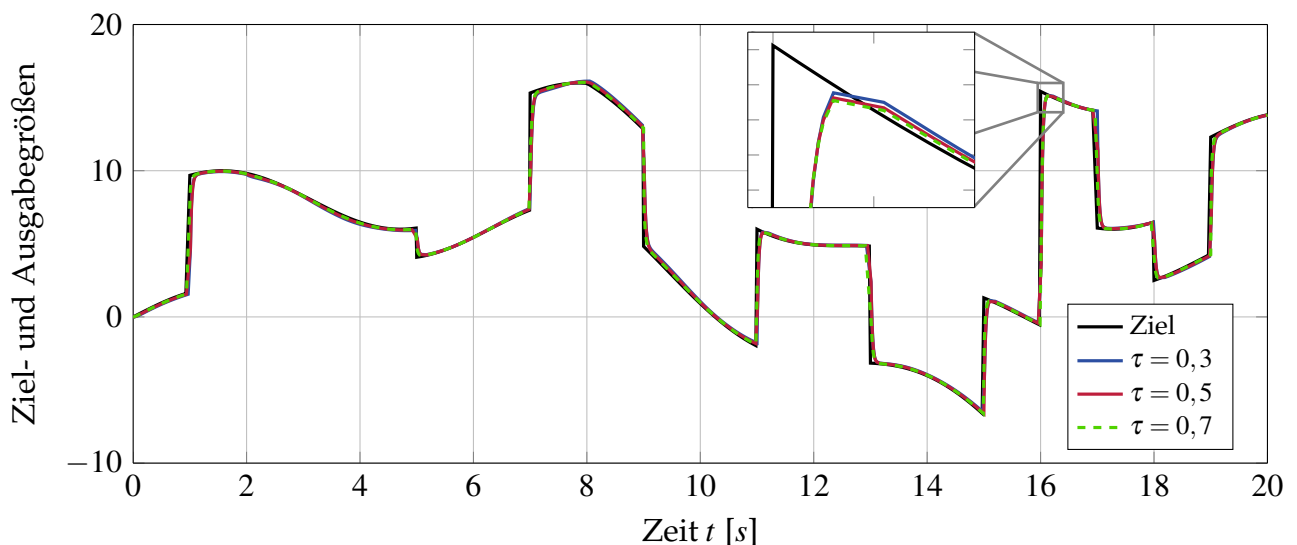


Abbildung 4.13: Vergleich der Netzwerkausgaben für einen komplexen Zielgrößenverlauf mit verschiedenen Intervallparametern und $\lambda = 50$ sowie σ^* -Korrektur

Es ist klar zu erkennen, dass die neuronalen Netzwerke mit jedem der drei verwendeten Intervallparametern in der Lage sind, sich schnell und stabil an die vorgegebene Zielgröße anzupassen. Zur Auswertung der schlecht zu differenzierenden Verläufe wird der für die Quantifizierung von Abbildungsfehlern übliche mittlere quadratische Fehler MSE (engl. Mean-Square-Error) nach Gleichung 4.42 verwendet.

$$\text{MSE} = \frac{1}{n} \cdot \sum_{i=1}^n (y_{k,i} - y_i)^2 = \frac{1}{n} \cdot \sum_{i=1}^n (\varepsilon_{t,i})^2 \quad (4.42)$$

Die Fehlerauswertung der in Abbildung 4.13 dargestellten Netzwerkausgänge kann Tabelle 4.4 entnommen werden, wobei das beste Trainingsergebnis grau hinterlegt ist. Zusätzlich sind die Mittelwerte der Lernratenbeträge $|\bar{\mu}|$ aufgelistet.

Intervallparameter τ	MSE	$ \bar{\mu} $
0,3	0,7540	0,0195
0,5	0,7379	0,0306
0,7	0,5532	0,0477

Tabelle 4.4: Fehlerauswertung bei unterschiedlichen Intervallparametern

Wie schon im ersten untersuchten Trainingsszenario, weist das Gleitzustandsverfahren mit $\tau = 0,7$ den geringsten Fehler über die simulierten 20 Sekunden auf. Dies ist auf eine erhöhte Konvergenzgeschwindigkeit als Konsequenz der im Mittel höheren Lernrate zurückzuführen. So lässt sich feststellen, dass die durch den Intervallparameter bedingte und durch den Konvergenzquotienten ausgedrückte schnellere Erreichung der Schaltlinie bei $\tau = 0,5$, auch im Falle des komplexen Trainingsszenarios, nicht direkt in einen schnelleren Ausgleich des Ausgabefehlers übersetzt wird. Stattdessen hat die Wahl eines größeren Intervallparameters die Erzeugung höherer Lernraten zur Folge, die letztendlich ein beschleunigten Ausgleich des Fehlers ermöglichen.

Zur Veranschaulichung der durch den Intervallparameter skalierten Lernraten ist deren zeitlicher Verlauf in Abbildung 4.14 dargestellt. Dabei ist nur der relevante Lernratenbereich dargestellt, so dass einzelne, nur einen Zeitschritt gültige, Lernraten nach oben und unten abgeschnitten sind. Anhand der Verläufe lässt sich gut die über die gesamte Simulationsdauer betragsmäßig größte Lernratenberechnung für $\tau = 0,7$ erkennen. Im Vergleich zur Darstellung der Lernraten für das erste Trainingsszenario in Abbildung 4.10 ist für das komplexere Training kein *chattering* der Lernraten zu erkennen. Dies ist durch die dauerhafte Änderung der Eingangssignale und der somit kontinuierlichen Erzeugung einer fehlerbehafteten Ausgangsgröße zu begründen, da auf diese Weise die Netzwerkzustände einen generell größeren Abstand zur Schaltlinie haben und es so nicht zum wiederholten Überschreiten dieser, als Ursache des *chatterings*, kommt.

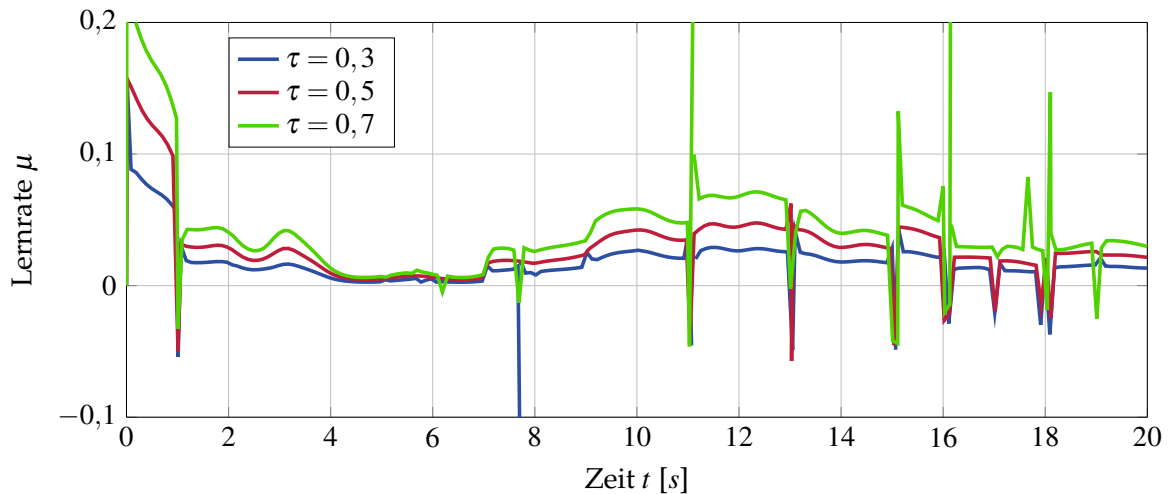


Abbildung 4.14: Vergleich der berechneten Lernraten mit $\lambda = 50$ sowie σ^* -Korrektur

Die Ergebnisse aus Tabelle 4.4 legen die Vermutung nahe, dass große τ zu einem immer weiter sinkenden MSE führen. Jedoch muss der für steigende τ ausführlich beschriebene Anstieg des Konvergenzquotienten bis hin zur Erreichung der oberen Stabilitätsgrenze und des folglich nicht mehr zu garantierenden Gleitzustands berücksichtigt werden. Als Beispiel des destabilisierenden Effekts der Lernratenberechnung bei Erreichung und Überschreitung der Intervallgrenzen sind in Abbildung 4.15 die Netzwerkausgaben für $\tau = 1$, $\tau = 1,3$ und $\tau = -0,3$ zu sehen. Für die gezeigten Verläufe wird keine σ^* -Korrektur verwendet, da im Falle der Nichterfüllung von Gleichung 4.38 der Intervallparameter auf $\tau = 0,5$ korrigiert würde und so der schädliche Effekt der Intervallgrenzenüberschreitung nicht sichtbar wäre.

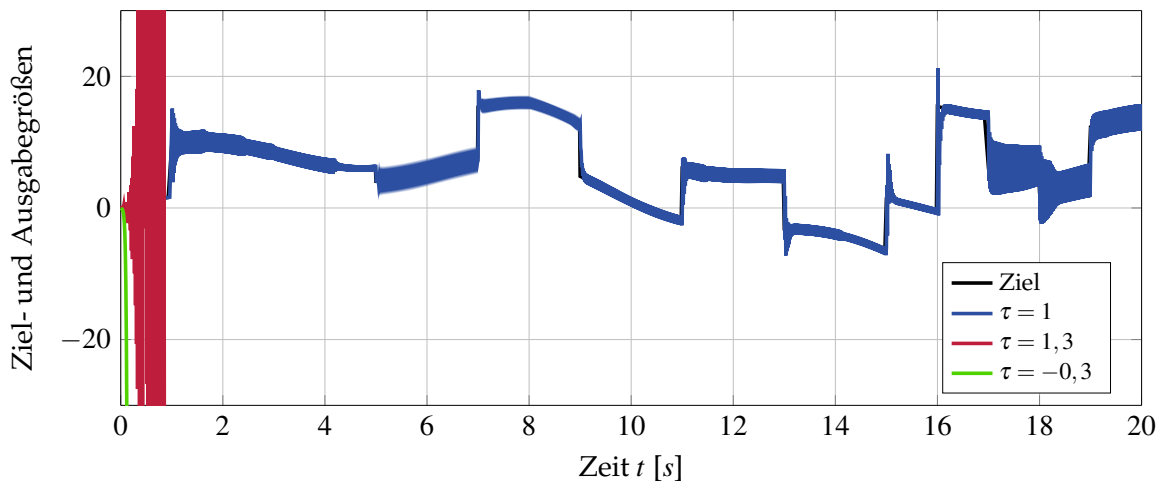


Abbildung 4.15: Vergleich der Netzwerkausgaben für einen komplexen Zielgrößenverlauf mit Erreichung und Überschreitung der Grenzen des Lernratenintervalls und $\lambda = 50$ ohne σ^* -Korrektur

Die gezeigten Verläufe bestätigen die im Rahmen der Herleitung des Lernratenintervalls aufgestellten Stabilitätsgrenzen bei $\tau = 0$ und $\tau = 1$. Für die dargestellte Netzausgabe mit dem Intervallparameter $\tau = 1$ lässt sich erkennen, dass das Training zwar noch grenzstabil

durchgeführt wird, die Ausgabe dabei jedoch dauerhaft um die Zielgröße oszilliert. Dies ist, wie in Tabelle 4.2 dargestellt, durch die nicht länger eintretende Konvergenz der Schaltfunktion und somit der Nichterreichung der Gleitlinie zu begründen. Bei der weiteren Erhöhung des Intervallparameters auf $\tau = 1,3$ vergrößert sich der Schaltfunktionsbetrag mit jedem Zeitschritt, so dass es folglich zur Instabilität des Trainings mit anhaltender Überschreitung der Gleitlinie kommt. Ebenfalls instabil verhält sich das Training bei der Überschreitung der unteren Stabilitätsgrenze mit $\tau = -0,3$. Dabei kommt es zu einem monotonen Anstieg des Schaltfunktionsbetrags ohne dabei die Gleitlinie zu überschreiten. Das Ergebnis ist ein kontinuierlich ansteigender Betrag der Netzausgabe.

Aus den dargestellten Gründen ist es zweckmäßig den Intervallparameter mit einer sinnvollen Reserve zu den Stabilitätsgrenzen zu wählen. Dabei ist es das Ziel, die Lernrate so hoch wie möglich zu bestimmen, ohne dass der dabei wachsende Konvergenzquotient $\frac{S_{t+1,i}}{S_{t,i}}$ zu übermäßigen Oszillationen der Netzwerkausgabe führt. Als Gütekriterium für die Bestimmung des optimalen Intervallparameters wird erneut der mittlere quadratische Ausgabebefehler MSE verwendet. Zur Berücksichtigung des Einflusses der Schaltfunktion auf die Wahl des Intervallparameters wird λ variiert, so dass jeweils der gesamte Wertebereich von τ für ein gewähltes λ simuliert wird. Der MSE für die Simulation des in Abbildung 4.12 gezeigten Verlaufs mit einem Wertebereichen von $\tau = [0 : 1]$ und $\lambda = [50 : 1000]$ ist in Abbildung 4.16 dargestellt.

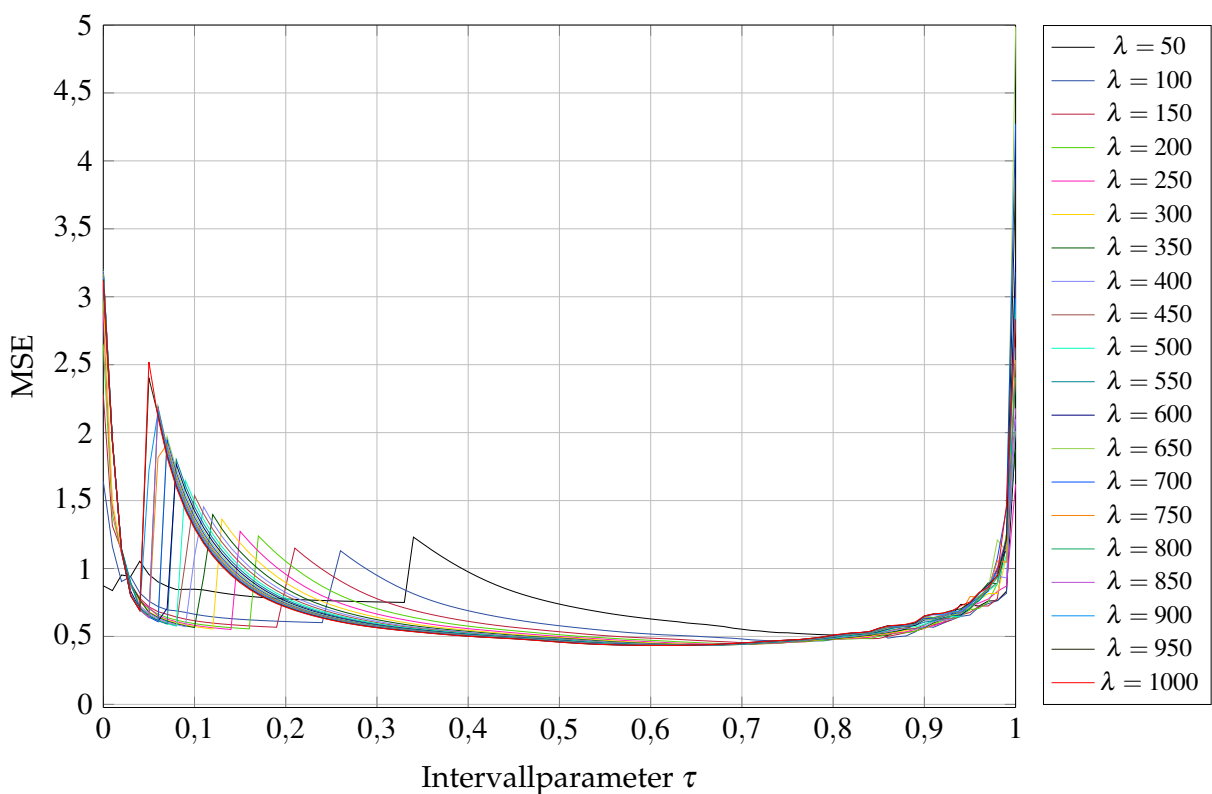


Abbildung 4.16: MSE Vergleich für verschiedene λ in Abhängigkeit des Intervallparameters τ für das Netzwerktraining mit σ^* -Korrektur

In den gezeigten Verläufen ist deutlich der Einfluss der σ^* -Korrektur in Form der sprunghaften MSE Änderung für $0 < \tau < 0,35$ zu sehen. In diesem Bereich wird, abhängig vom simulierten λ , der mittlere quadratische Fehler sprunghaft reduziert, was auf die gezielte Änderung der Lernratenbestimmung auf $\mu = \frac{a_i}{b_i}$ als Effekt der σ^* -Korrektur zurückzuführen ist. Dabei ist der Wertebereich, in dem die σ^* -Korrektur nötig ist, umso kleiner, je größer λ gewählt wird. Die entsprechenden MSE-Verläufe für die Simulationen ohne σ^* -Korrektur sind in Anhang A.2 dargestellt. Generell kann für beide Darstellungen festgehalten werden, dass der mittlere quadratische Fehler in der Nähe der Stabilitätsgrenzen am höchsten ist und zur Mitte des Lernratenintervalls hin abnimmt. Dabei zeigt sich eine klare Abhängigkeit von dem gewählten λ , die für steigende Werte des Schalfunktionsparameters jedoch abnimmt. Um dieses Verhalten detaillierter darzustellen, ist in Abbildung 4.17 ein vergrößerter Ausschnitt der vorherigen Darstellung zu sehen.

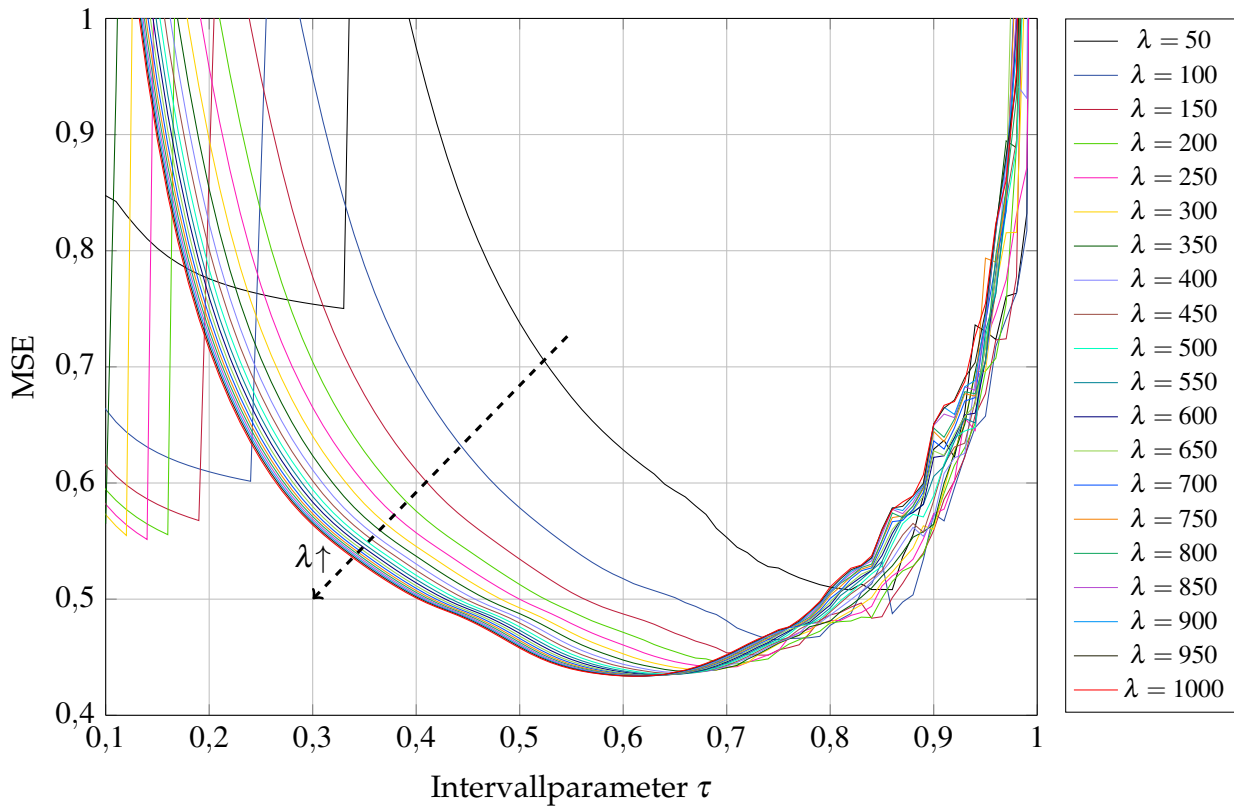


Abbildung 4.17: Ausschnitt des MSE Vergleichs für verschiedene λ in Abhängigkeit des Intervallparameters τ für das Netzwerktraining mit σ^* -Korrektur

Anhand dieser Abbildung ist gut die abnehmende Tendenz des mittleren quadratischen Fehlers für τ im Bereich von 0,6 bis 0,7 zu sehen. Dabei ist zunächst ein großer Unterschied zwischen $\lambda = 50$ und beispielsweise $\lambda = 300$ erkennbar, wohingegen die weitere Erhöhung des Schalfunktionsparameters nur noch geringe Auswirkung auf den MSE hat. Ebenfalls eindeutig zu erkennen sind die beginnenden Oszillationen bei der Annäherung an die obere Grenze bei $\tau = 1$. So ist ab einem Intervallparameter von ungefähr 0,8 eine hohe Varianz des Fehlers zu sehen, was im Kontrast zu dem ansonsten gleichmäßig ab- und zunehmen-

den MSE-Verlauf steht. So legen diese Ergebnisse eine Stabilitätsreserve von mindestens 20% zur oberen Lernratenbegrenzung nahe. Ein entsprechendes Verhalten kann für die Simulation ohne σ^* -Korrektur in Anhang A.3 gesehen werden. Über das absolute Minimum des mittleren quadratischen Fehlers hinaus, ist in Abbildung 4.17 auch die Tendenz des optimalen τ für ein gewähltes λ zu erkennen. Zur Verdeutlichung dieses Verhaltens ist in Abbildung 4.18 eine farblich kodierte Oberflächendarstellung des mittleren quadratischen Fehlers in Abhängigkeit von τ und λ dargestellt, wobei jeder $\text{MSE} > 0,7$ pauschal als dunkelrot abgebildet ist. Die eingefügte graue Linie verbindet dabei die jeweiligen Bereiche mit minimalem MSE für ein gegebenes λ . Auf diese Weise ist gut die abnehmende Tendenz des optimalen Intervallparameters für steigende λ abzulesen.

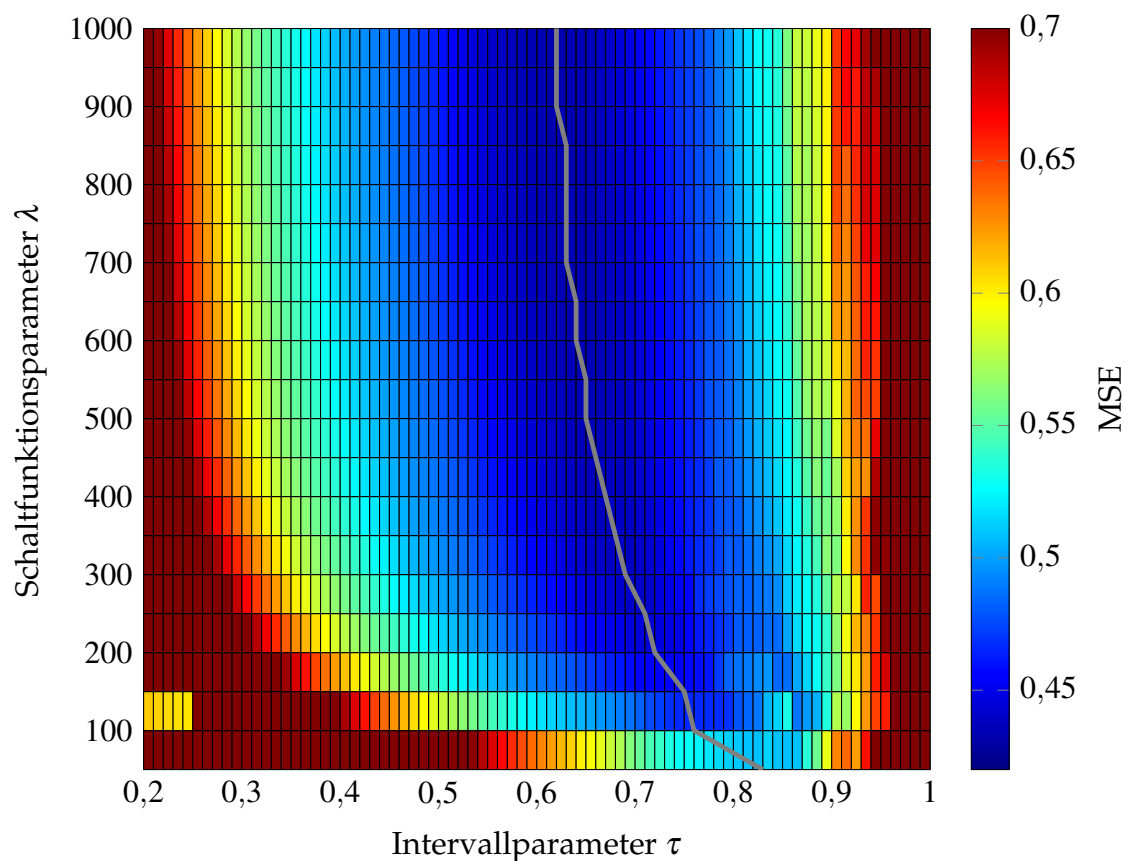


Abbildung 4.18: Oberflächendarstellung des mittleren quadratischen Fehlers für verschiedene λ in Abhängigkeit des Intervallparameters τ für das Netzwerktraining mit σ^* -Korrektur

Aus sowohl Abbildung 4.17 als auch aus Abbildung 4.18 geht hervor, dass verschiedene Kombinationen von λ und τ zu sehr ähnlichen Ergebnissen in Bezug auf den mittleren quadratischen Fehler führen. So ergeben sich beispielsweise für die Parameterkombinationen von $\lambda = 800$ und $\tau = 0,63$ sowie $\lambda = 300$ und $\tau = 0,69$ nahezu übereinstimmende mittlere quadratische Fehler, was in einem Ausschnitt des 20 Sekunden langen Trainingsszenarios in Abbildung 4.19 dargestellt ist.

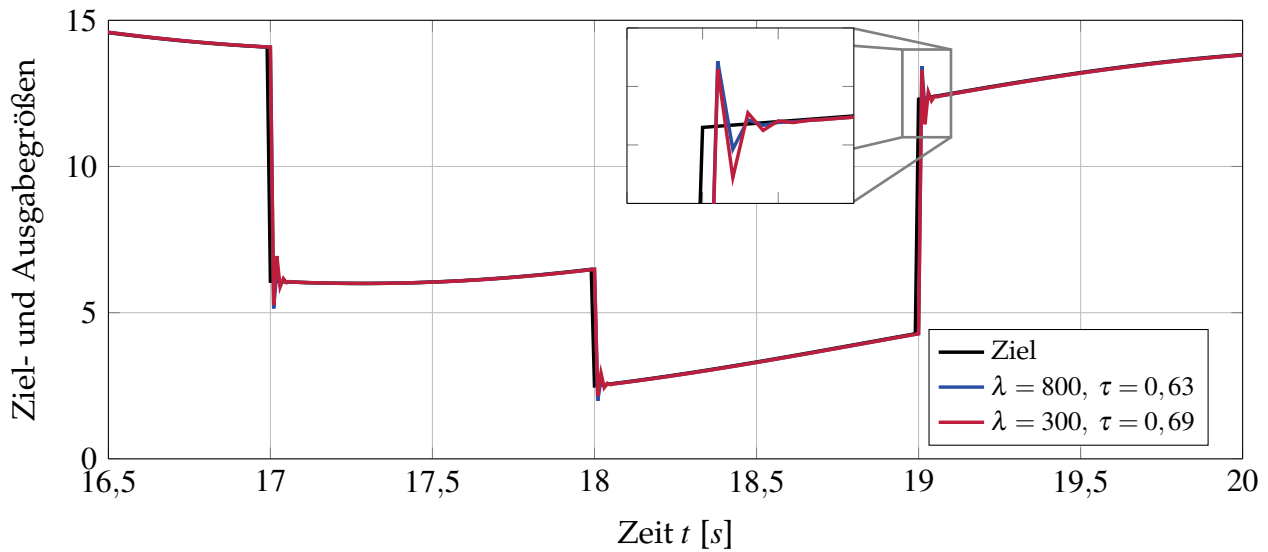


Abbildung 4.19: Vergleich der Netzwerkausgaben für zwei verschiedene Kombinationen von λ und τ

Gut zu sehen ist die Tendenz zum Über- und Unterschneiden der Zielgröße als direkte Reaktion auf die Sprünge. Wie oben dargestellt, wird dieses Verhalten durch die Wahl des Intervallparameters τ maßgeblich beeinflusst, da dieser bei $\tau > 0,5$ für das Überschreiten der Gleitlinie im Phasenraum verantwortlich ist und somit das zunächst schwingende Verhalten der Netzausgabe begünstigt. Für den Vergleich des Gleitzustandtrainings mit dem klassischen Backpropagation-Algorithmus für das hier gezeigte Trainingsszenario werden zwei unterschiedliche Parameterkombinationen von λ und τ verwendet, wie sie in Tabelle 4.5 mit dazu gehörigen MSE des komplexen Trainingsszenarios aufgelistet sind. Dabei steht der Name *SMC-FO* für Sliding-Mode-Control-First-Order.

Name	λ	τ	MSE
SMC-FO1	800	0,45	0,49
SMC-FO2	300	0,65	0,44

Tabelle 4.5: Parameterkombination für das Gleitzustandtraining erster Ordnung (Sliding-Mode-Control-First-Order)

Für *SMC-FO1* wird ein Intervallparameter kleiner 0,5 gewählt, um auf diese Weise ein Überschreiten der Gleitlinie im Phasenraum zu verhindern und so trotz des hohen λ von 800 einen schwingungsfreien Netzwerkausgang zu gewährleisten. Für die zweite Kombination *SMC-FO2* wird im Vergleich zu Abbildung 4.19 ein leicht verkleinerter Intervallparameter gewählt, um die Tendenz zum Überschwingen geringfügig zu verringern, dabei jedoch möglichst nahe am für dieses Trainingsszenario spezifischen simulierten Minimum des MSE zu bleiben. Zum Vergleich der Leistungsfähigkeit der auf diese Weise trainierten Netzwerke für das dargestellte komplexe Trainingsszenario, sind in Abbildung 4.20 die Netzwerkaus-

gaben für den oben gezeigten Ausschnitt dargestellt. Dabei wird als Referenz der konventionelle Backpropagation-Algorithmus (*BP*) mit konstanter Lernrate verwendet. Für die Wahl der Lernrate wird $\mu = 0,03$ angenommen, da es sich dabei um den maximalen Wert handelt, bei dem es nicht zu spontanen Oszillationen als Reaktion auf eine ungünstige Konstellation aus Eingangssignalen und Zielwerten mit einer zu hohen Lernrate kommt.

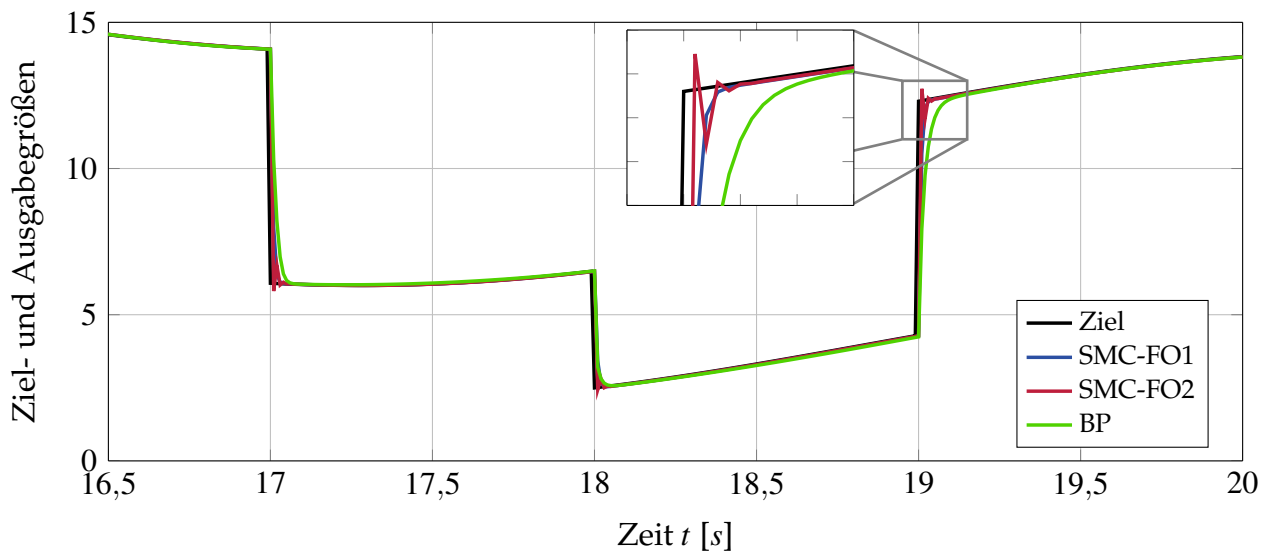


Abbildung 4.20: Vergleich der zwei Parameterkombinationen des Gleitzustandstrainings erster Ordnung mit dem Backpropagation-Algorithmus

Durch die Wahl einer konstanten Lernrate limitiert, zeigt das backpropagation Training das schlechteste Trainingsergebnis. Die konservative Lernratenauslegung zum Schutz vor ungewünschten Oszillationen der Netzausgabe führt zu einer deutlich erkennbaren Reduktion der Konvergenzgeschwindigkeit im direkten Vergleich mit den beiden Gleitzustandsverfahren. Es zeigt sich, dass das mit $\tau = 0,45$ trainierte *SMC-FO1*, wie aus der Theorie abgeleitet, trotz eines großen Schaltfunktionsparameters von $\lambda = 800$ nicht zum Überschießen der Trainingsgröße neigt, dabei jedoch eine leicht reduzierte Konvergenzgeschwindigkeit gegenüber *SMC-FO2* aufweist.

Ebenfalls vom Verlauf direkt aus der Herleitung der Lernratenberechnung ableitbar, sind die in Abbildung 4.21 dargestellten Phasenportraits im Zustandsraum. Der Intervallparameter $\tau = 0,45$ von *SMC-FO1* führt auch im komplexen Trainingsszenario dazu, dass die Gleitlinie, mit einer einzigen Ausnahme, nicht von der Zustandstrajektorie überschritten wird. So ist gut die tangentielle Annäherung an die gestrichelt dargestellte Gleitlinie zu erkennen. Dieser mit der Theorie übereinstimmende Verlauf unterstreicht die Fähigkeit des vorgestellten Lernverfahrens, die Transparenz des Trainingsprozesses zu erhöhen, indem die Trajektorie des Ausgabefehlers ε im Phasenraum vorgegeben werden kann.

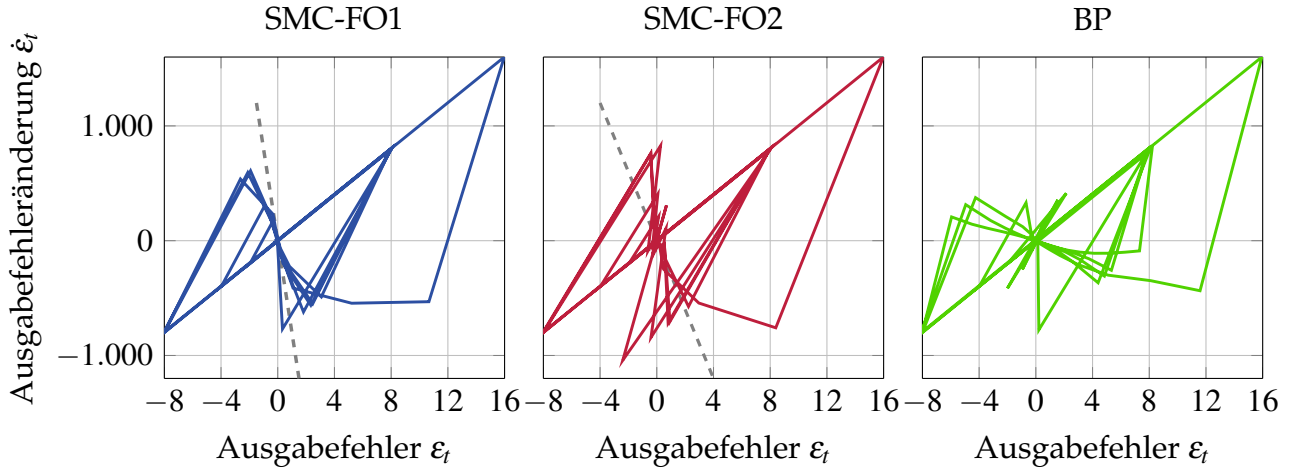


Abbildung 4.21: Phasenportraits der unterschiedlichen Lernverfahren für das komplexe Trainingsszenario

Auch für das zweite Gleitzustandstraining *SMC-FO2* ist das oben hergeleitete Verhalten in Form der Überschreitung der Gleitlinie durch $\tau = 0,65$ gut zu erkennen. Dieses führt in einigen Fällen ebenfalls zum Überschreiten von $\epsilon = 0$ und somit zum Überschießen der Zielgröße, was in Abbildung 4.20 als Überschwinger zu sehen ist. Dem Phasenportrait kann direkt entnommen werden, dass die Überschwinger das Resultat der Überschreitung der Gleitlinie in Kombination mit der gewählten Steigung λ sind. Das Phasenportrait des Backpropagation-Algorithmus zeigt eine weniger gerichtete Zustandstrajektorie, mit teils großen Änderungen des Fehlers innerhalb eines Zeitschritts und teils nur sehr langsamer Annäherungen an den stabilen Ursprung. Dies verdeutlicht die Charakteristik des Gradientenabstiegsverfahrens mit fester Schrittweite nur ungenügend auf die Größe und Änderung des Fehlers reagieren zu können, so dass es sowohl zu einer langsameren Konvergenz als auch zu Stabilitätsproblemen bei zu großen Fehlern kommen kann.

4.4 Gleitzustandstraining zweiter Ordnung

Die Umsetzung des Gleitzustandstrainings zweiter Ordnung in Anlehnung an den Twisting-Algorithmus wird, analog dem oben dargestellten Ansatz, über die Berechnung einer Lernrate μ realisiert, die zur Erreichung und Beibehaltung des Gleitzustands führt. Dabei wird erneut der bereits für das Verfahren erster Ordnung verwendete Ausdruck für die Gewichtsänderung

$$\Delta \mathbf{w} = \left(\frac{\partial \vec{y}(\mathbf{w}, \vec{x})}{\partial \mathbf{w}} \right)^T \cdot \mu_{Tw} \cdot \text{diag} \left[\text{sign}(\vec{S}_t) \right] \cdot |\vec{\epsilon}| \quad (4.43)$$

genutzt. Als Erreichbarkeits- und Existenzbedingung wird dabei die Erweiterung der diskreten Kontraktionsbedingung $|S_{t+1}| < |S_t|$ um die betragsmäßige Reduktion von \vec{S}_t genutzt, was dem Prinzip des Twisting-Algorithmus entspricht. Wie bereits erwähnt, finden sich

Untersuchungen zu der alternativen Verwendung von *reaching laws* zweiter Ordnung als Erreichbarkeitsbedingung in [88–90].

4.4.1 Berechnung der Lernrate mit erweiterter Kontraktionsbedingung

Soll neben der Bedingung, dass S_t in jedem Zeitschritt betragsmäßig abnimmt, noch die erste Ableitung der Schaltfunktion für das Netzwerktraining genutzt werden, kann die Berechnung der Lernrate μ erweitert werden. Zu diesem Zweck wird eine aus der Umschaltung sowohl an S als auch an \dot{S} kombinierte Lernrate μ_{Tw} eingeführt. In Anlehnung an den Twisting-Algorithmus gilt für diese abgewandelte Lernratenbestimmung zweiter Ordnung:

$$\mu_{Tw} = \mu + \mu_{\dot{S}}. \quad (4.44)$$

Dabei handelt es sich bei μ um das Ergebnis der Lernratenbestimmung mit diskreter Erreichbarkeitsbedingung aus Gleichung 4.29 und bei $\mu_{\dot{S}}$ um den im Anschluss eingeführten Anteil mit Umschaltung an \dot{S} . Für das Gleitzustandstraining zweiter Ordnung mit erweiterter Kontraktionsbedingung soll neben der Bedingung aus Gleichung 4.5 zusätzlich

$$|\dot{S}_{t+1}| < |\dot{S}_t| \quad (4.45)$$

gelten. Mit Hilfe des Differenzenquotienten und der Annahme kleiner Schrittweiten T_s kann dieser Ausdruck wie folgt umgeschrieben werden:

$$\left| \frac{S_{t+1} - S_t}{T_s} \right| < \left| \frac{S_t - S_{t-1}}{T_s} \right|. \quad (4.46)$$

An dieser Stelle kann nun die in Abschnitt 4.3.2 hergeleitete Näherung $S_{t+1,i} = a_i - \mu \cdot b_i$ aus Gleichung 4.17 eingesetzt werden. Dabei steht der Index i erneut für das betrachtete Ausgabeneuron. In Abhängigkeit von $\mu_{\dot{S}}$ gesetzt, ergibt sich für die Kontraktionsbedingung somit:

$$|\dot{S}_{t+1}| < |\dot{S}_t| \Rightarrow \left| \frac{a_i - \mu_{\dot{S}} \cdot b_i - S_t}{T_s} \right| < \left| \frac{S_t - S_{t-1}}{T_s} \right|. \quad (4.47)$$

Wie auch für das Gleitzustandstraining erster Ordnung können bei der Auflösung der beiden Beträge in Gleichung 4.47 wieder vier Fälle unterschieden werden. In Tabelle 4.6 sind diese in zwei Gruppen sortiert aufgeführt, abhängig davon, ob das Vorzeichen von $\dot{S}_{t,i}$ im kommenden Zeitschritt wechselt.

Fall	Betrag 1	Betrag 2
Fall 1a:	$\dot{S}_{t+1,i} > 0$	$\dot{S}_{t,i} > 0$
Fall 1b:	$\dot{S}_{t+1,i} < 0$	$\dot{S}_{t,i} < 0$
Fall 2a:	$\dot{S}_{t+1,i} < 0$	$\dot{S}_{t,i} > 0$
Fall 2b:	$\dot{S}_{t+1,i} > 0$	$\dot{S}_{t,i} < 0$

Tabelle 4.6: Fallunterscheidung der zeitdiskreten Kontraktionsbedingung für $\dot{S}_{t,i}$

Fall 1a: $\dot{S}_{t+1,i} > 0$, $\dot{S}_{t,i} > 0$ mit $b_i > 0$

Für den betrachteten Fall zweier positiver Beträge und Multiplikation mit T_s folgt für die Kontraktionsbedingung:

$$a_i - \mu_{\dot{S}} \cdot b_i - S_{t,i} < S_{t,i} - S_{t-1,i} . \quad (4.48)$$

Für das weitere Auflösen nach $\mu_{\dot{S}}$ muss an dieser Stelle durch b_i geteilt werden. Dabei kann, im Gegensatz zu der Fallunterscheidung für das Gleitzustandstraining erster Ordnung, keine pauschale Aussage über das Vorzeichen dieser Ersatzgröße getroffen werden. Es ist somit notwendig, die beiden Fälle $b_i < 0$ und $b_i > 0$ getrennt zu betrachten. Allerdings wird an dieser Stelle schnell ersichtlich, dass sich durch das Vorzeichen von b_i lediglich die Richtung des Ungleichheitszeichen umkehrt, der formelmäßige Ausdruck jedoch unverändert bleibt. So wird aus Gründen der Übersichtlichkeit an dieser Stelle die ausführliche Fallunterscheidung für $b_i > 0$ dargestellt und bei der Zusammenfassung aller vier Fälle das Ergebnis für $b_i < 0$ mit getauschten Ungleichheitszeichen mit aufgeführt. Für die untere Grenze der Lernratenberechnung mit Schaltvorgang an $\dot{S}_{t,i}$ folgt somit:

$$\mu_{\dot{S}} > \frac{a_i + S_{t-1,i} - 2 \cdot S_{t,i}}{b_i} . \quad (4.49)$$

Zum Aufstellen einer oberen Grenze wird sich die für **Fall 1a** angenommene Bedingung $\dot{S}_{t+1,i} > 0$ zu Nutze gemacht. Unter Verwendung des Differenzenquotienten folgt daraus direkt, dass $S_{t+1,i} > S_{t,i}$ gelten muss. An dieser Stelle, erneut nur für $b_i > 0$ dargestellt, ergibt sich:

$$a_i - \mu_{\dot{S}} \cdot b_i > S_{t,i} \Rightarrow \mu_{\dot{S}} < \frac{a_i - S_{t,i}}{b_i} . \quad (4.50)$$

Die obere Grenze aus Gleichung 4.50 wird, wie von dem Gleitzustandstraining erster Ordnung bekannt, zu $\mu_{\dot{S}} < \frac{a_i}{b_i}$ sobald der Gleitzustand bei $S_{t,i} = 0$ erreicht wurde. Anders verhält es sich für die untere Grenze aus Gleichung 4.49, für die zwei aufeinander folgende Zeitschritte im Gleitzustand notwendig sind, bis $\mu_{\dot{S}} > \frac{a_i}{b_i}$ gilt. Es ist somit zu erkennen, dass die in $\dot{S}_{t,i}$ transportierte Information über die Änderung der Schaltfunktion in **Fall 1a** durch die untere Lernratenbegrenzung umgesetzt wird.

Fall 1b: $\dot{S}_{t+1,i} < 0$, $\dot{S}_{t,i} < 0$ mit $b_i > 0$

Für den zweiten betrachteten Fall mit zwei negativen Beträgen werden beide Seiten der Ungleichung mit minus Eins multipliziert.

$$-a_i + \mu_{\dot{S}} \cdot b_i + S_{t,i} < -S_{t,i} + S_{t-1,i} \quad (4.51)$$

Für die an dieser Stelle erneut nur mit positivem Vorzeichen von b_i dargestellte Herleitung folgt:

$$\mu_{\dot{S}} < \frac{a_i + S_{t-1,i} - 2 \cdot S_{t,i}}{b_i} . \quad (4.52)$$

Dabei handelt es sich um die obere Begrenzung der Lernrate $\mu_{\dot{S}}$, die der unteren Begrenzung aus **Fall 1a** entspricht. Aufgrund der getroffenen Annahme $\dot{S}_{t+1,i} < 0$ folgt aus dem Differenzenquotienten direkt $S_{t+1,i} < S_{t,i}$. Für die untere Begrenzung der Lernrate kann somit geschrieben werden:

$$a_i - \mu_{\dot{S}} \cdot b_i < S_{t,i} \Rightarrow \mu_{\dot{S}} > \frac{a_i - S_{t,i}}{b_i} . \quad (4.53)$$

Damit ist im **Fall 1b** mit $b_i > 0$ nur die obere Grenze von der Schaltfunktionsänderung $\dot{S}_{t,i}$ abhängig, da in der unteren Begrenzung aus Gleichung 4.53 nicht die Schaltfunktion aus dem letzten Zeitschritt $S_{t-1,i}$ und somit keine Gradienteninformation auftaucht.

Fall 2a: $\dot{S}_{t+1,i} < 0$, $\dot{S}_{t,i} > 0$ mit $b_i > 0$

In diesem Fall ändert sich das Vorzeichen der Schaltfunktionsableitung von einem positiven Betrag im aktuellen zu einem negativen Betrag im nächsten Zeitschritt. Für die Kontraktionsbedingung mit aufgelösten Beträgen und angepasster linken Seite folgt:

$$-a_i + \mu_{\dot{S}} \cdot b_i + S_{t,i} < S_{t,i} - S_{t-1,i} . \quad (4.54)$$

Für ein positives b_i folgt für **Fall 2a** eine obere Begrenzung der Lernrate in der Form:

$$\mu_{\dot{S}} < \frac{a_i - S_{t-1,i}}{b_i} . \quad (4.55)$$

Auffällig dabei ist, dass in diesem Fall die Lernratenbegrenzung nicht mehr vom aktuellen sondern ausschließlich vom vorangegangenen Zeitschritt der Schaltfunktion abhängt. Wie auch in **Fall 1b** folgt aus $\dot{S}_{t+1,i} < 0$ direkt die obere Begrenzung für $\mu_{\dot{S}}$ zu:

$$a_i - \mu_{\dot{S}} \cdot b_i < S_{t,i} \Rightarrow \mu_{\dot{S}} > \frac{a_i - S_{t,i}}{b_i} . \quad (4.56)$$

Fall 2b: $\dot{S}_{t+1,i} > 0$, $\dot{S}_{t,i} < 0$ mit $b_i > 0$

Für den letzten betrachteten Fall ändert sich der positive Betrag der Schaltfunktionsänderung im aktuellen Zeitschritt zu einem negativen Wert im kommenden Zeitschritt. Die daraus resultierende Ungleichung mit aufgelösten Betragsklammern ergibt sich zu:

$$a_i - \mu_{\dot{S}} \cdot b_i - S_{t,i} < -S_{t,i} + S_{t-1,i} . \quad (4.57)$$

Mit an dieser Stelle wiederum nur betrachtetem $b_i > 0$ folgt für die untere Begrenzung der Lernrate:

$$\mu_{\dot{S}} > \frac{a_i - S_{t-1,i}}{b_i} . \quad (4.58)$$

Entsprechend der Herleitung in **Fall 1a** kann aus der Bedingung $\dot{S}_{t+1,i} > 0$ eine obere Grenze berechnet werden. Es folgt für positive b_i :

$$a_i - \mu_{\dot{S}} \cdot b_i > S_{t,i} \Rightarrow \mu_{\dot{S}} < \frac{a_i - S_{t,i}}{b_i} . \quad (4.59)$$

Damit wurden für alle vier Fälle unter der Annahme $b_i > 0$ sowohl eine obere als auch eine untere Grenze gefunden.

Für eine übersichtlichere Auflistung der aufgestellten Grenzen unter Berücksichtigung des Vorzeichens von b_i werden jeweils die Fälle mit gleichem Vorzeichen von $\dot{S}_{t,i}$ zusammengefasst. Somit folgt für **Fall 1a** und **Fall 2a**:

$$\left\{ \begin{array}{l} \left\{ \mu_{\dot{S}} > \frac{a_i + S_{t-1,i} - 2 \cdot S_{t,i}}{b_i} \wedge \mu_{\dot{S}} < \frac{a_i - S_{t,i}}{b_i} \right\} \vee \left\{ \mu_{\dot{S}} > \frac{a_i - S_{t,i}}{b_i} \wedge \mu_{\dot{S}} < \frac{a_i - S_{t-1,i}}{b_i} \right\} \text{ für } b_i > 0 \\ \left\{ \mu_{\dot{S}} < \frac{a_i + S_{t-1,i} - 2 \cdot S_{t,i}}{b_i} \wedge \mu_{\dot{S}} > \frac{a_i - S_{t,i}}{b_i} \right\} \vee \left\{ \mu_{\dot{S}} < \frac{a_i - S_{t,i}}{b_i} \wedge \mu_{\dot{S}} > \frac{a_i - S_{t-1,i}}{b_i} \right\} \text{ für } b_i < 0 . \end{array} \right. \quad (4.60)$$

Dem Vorgehen für das Gleitzustandstraining erster Ordnung entsprechend, können die jeweiligen Intervalle für ein Vorzeichen von b_i in einem Vereinigungsintervall zusammengefasst werden.

$$\left\{ \begin{array}{l} \mathcal{L} = \left\{ \mu_{\dot{S}} \in \mathfrak{R} \mid \frac{a_i + S_{t-1,i} - 2 \cdot S_{t,i}}{b_i} < \mu_{\dot{S}} < \frac{a_i - S_{t-1,i}}{b_i} \right\} \text{ für } b_i > 0 \\ \mathcal{L} = \left\{ \mu_{\dot{S}} \in \mathfrak{R} \mid \frac{a_i - S_{t-1,i}}{b_i} < \mu_{\dot{S}} < \frac{a_i + S_{t-1,i} - 2 \cdot S_{t,i}}{b_i} \right\} \text{ für } b_i < 0 \end{array} \right. \quad (4.61)$$

Auf die gleiche Weise können nun die Grenzen für **Fall 1b** und **Fall 2b** unter Berücksichtigung des Vorzeichens von b_i zusammengefasst werden. Es folgt:

$$\left\{ \begin{array}{l} \left\{ \mu_{\dot{S}} < \frac{a_i + S_{t-1,i} - 2 \cdot S_{t,i}}{b_i} \wedge \mu_{\dot{S}} > \frac{a_i - S_{t,i}}{b_i} \right\} \vee \left\{ \mu_{\dot{S}} < \frac{a_i - S_{t,i}}{b_i} \wedge \mu_{\dot{S}} > \frac{a_i - S_{t-1,i}}{b_i} \right\} \text{ für } b_i > 0 \\ \left\{ \mu_{\dot{S}} > \frac{a_i + S_{t-1,i} - 2 \cdot S_{t,i}}{b_i} \wedge \mu_{\dot{S}} < \frac{a_i - S_{t,i}}{b_i} \right\} \vee \left\{ \mu_{\dot{S}} > \frac{a_i - S_{t,i}}{b_i} \wedge \mu_{\dot{S}} < \frac{a_i - S_{t-1,i}}{b_i} \right\} \text{ für } b_i < 0 . \end{array} \right. \quad (4.62)$$

Dabei fällt auf, dass es sich bei den in Gleichung 4.62 aufgeführten Intervallen für $b_i > 0$ um die identischen Grenzen handelt, wie sie schon für den Fall mit $\dot{S}_{t,i} > 0$ und $b_i < 0$ aufgestellt wurden. Aus diesem Grund können die vier betrachteten Fälle weiter zusammengefasst und in zwei alle Vorzeichenkombinationen von $\dot{S}_{t,i}$ und b_i abdeckende Intervalle vereinigt werden. Dabei werden erneut die Kontraktionsbedingungen gelockert, so dass auch Gleichheit anstatt strikter Ungleichheit betrachtet wird.

$$\left\{ \begin{array}{l} \mathcal{L} = \left\{ \mu_{\dot{S}} \in \mathfrak{R} \mid \frac{a_i + S_{t-1,i} - 2 \cdot S_{t,i}}{b_i} \leq \mu_{\dot{S}} \leq \frac{a_i - S_{t-1,i}}{b_i} \right\} \text{ für } (\dot{S}_{t,i} \geq 0 \wedge b_i > 0) \vee (\dot{S}_{t,i} < 0 \wedge b_i < 0) \\ \mathcal{L} = \left\{ \mu_{\dot{S}} \in \mathfrak{R} \mid \frac{a_i - S_{t-1,i}}{b_i} \leq \mu_{\dot{S}} \leq \frac{a_i + S_{t-1,i} - 2 \cdot S_{t,i}}{b_i} \right\} \text{ für } (\dot{S}_{t,i} \geq 0 \wedge b_i < 0) \vee (\dot{S}_{t,i} < 0 \wedge b_i > 0) \end{array} \right. \quad (4.63)$$

Zur Veranschaulichung des Gleitzustandstrainings zweiter Ordnung mit zwei diskreten Kontraktionsbedingungen und einer erweiterter Lernratenbestimmung ist in Abbildung 4.22 das Blockschaltbild für ein Netzwerk mit einer Ausgangsgröße dargestellt. Mit den in Gleichung 4.63 aufgestellten Grenzen ist es nun also möglich, die an $\dot{S}_{t,i}$ schaltende Lernrate $\mu_{\dot{S}}$ zu berechnen, mit deren Hilfe die Lernrate μ des Gleitzustandstrainings erster Ordnung erweitert werden kann. Zur Bestimmung von $\mu_{\dot{S}}$ innerhalb des aufgespannten Intervalls, wird ein zweiter Intervallparameter $\tau_{\dot{S}}$ verwendet. Dabei ist es möglich, einen einheitlichen Wert für sowohl τ als auch $\tau_{\dot{S}}$ oder zwei sich unterscheidende Parameter zu wählen.

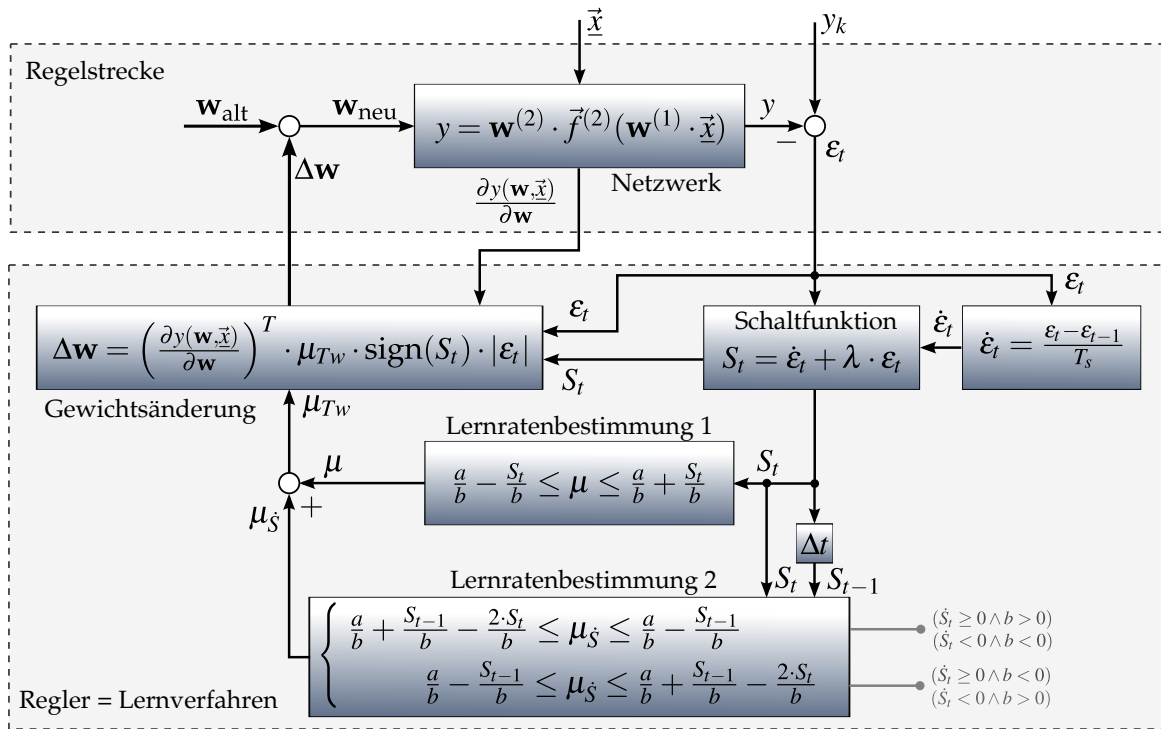


Abbildung 4.22: Blockschaltbild des Gleitzustandstrainings zweiter Ordnung für ein Netzwerk mit einer Ausgangsgröße

Wichtig zu beachten ist die durch $\mu_{\dot{S}}$ mögliche Erhöhung der kombinierten Lernrate μ_{Tw} über die in Abschnitt 4.3.2 aufgestellten Grenzen hinaus. Um weiterhin stabile Konvergenz der Zustände zu garantieren, kann die kombinierte Lernrate μ_{Tw} durch das für das Gleitzustandstraining erster Ordnung aufgestellte Intervall begrenzt werden. So ist sichergestellt, dass trotz der Erweiterung zum Gleitzustandstraining zweiter Ordnung, die Lernrate zu jeder Zeit innerhalb stabiler Grenzen berechnet wird. Dabei wird die in Abschnitt 4.3.3 vorgeschlagene Stabilitätsreserve von 20% zur oberen und unteren Intervallsgrenze ebenfalls für das Gleitzustandstraining zweiter Ordnung verwendet. Im folgenden Abschnitt wird der Einfluss der Lernratenerweiterung auf das Netzwerktraining untersucht und das Gleitzustandstraining zweiter Ordnung mit dem Ansatz erster Ordnung verglichen.

4.4.2 Untersuchung des Einflusses von $\tau_{\dot{S}}$ auf das Netzwerktraining

Für die Untersuchungen des Ansatzes zweiter Ordnung wird erneut das neuronale Netzwerk mit 5 Eingabe-, 15 verdeckten und einem Ausgabeneuron aus Abschnitt 4.3.3 betrachtet. Um zunächst die generelle Übertragbarkeit der Gleitzustandsregelung zweiter Ordnung auf das Training neuronaler Netzwerke zu demonstrieren, wird das einfache Trainingsszenario mit zweifachem Sollgrößensprung und konstanten Eingangssignalen aus Abbildung 4.7 untersucht. Dabei werden die Intervallparameter auf einen einheitlichen Wert in der Mitte des jeweiligen Lernratenintervalls mit $\tau = 0,5$ und $\tau_{\dot{S}} = 0,5$ gesetzt. Für den Schaltsfunktionsparameter wird der Wert $\lambda = 50$ angenommen. Zusätzlich wird in dieser ersten Analyse weder eine σ^* -Korrektur des Lernratenvorzeichens vorgenommen, noch

die kombinierte Lernrate μ_{Tw} zusätzlich durch die Stabilitätsgrenzen des Gleitzustandsverfahrens erste Ordnung beschränkt. Auf diese Weise soll das unmodifizierte Verhalten der Übertragung des Twisting-Algorithmus auf das Training neuronaler Netzwerke untersucht werden. In Abbildung 4.23 ist das Phasenportrait des beschriebenen Netzwerks für das Gleitzustandstraining zweiter Ordnung dargestellt.

Es sind deutlich die für den Twisting-Algorithmus charakteristischen wirbelartigen Zustandsverläufe nach den Sprüngen auf $y_k = 8$ beziehungsweise $y_k = 4$ zu erkennen. Dabei

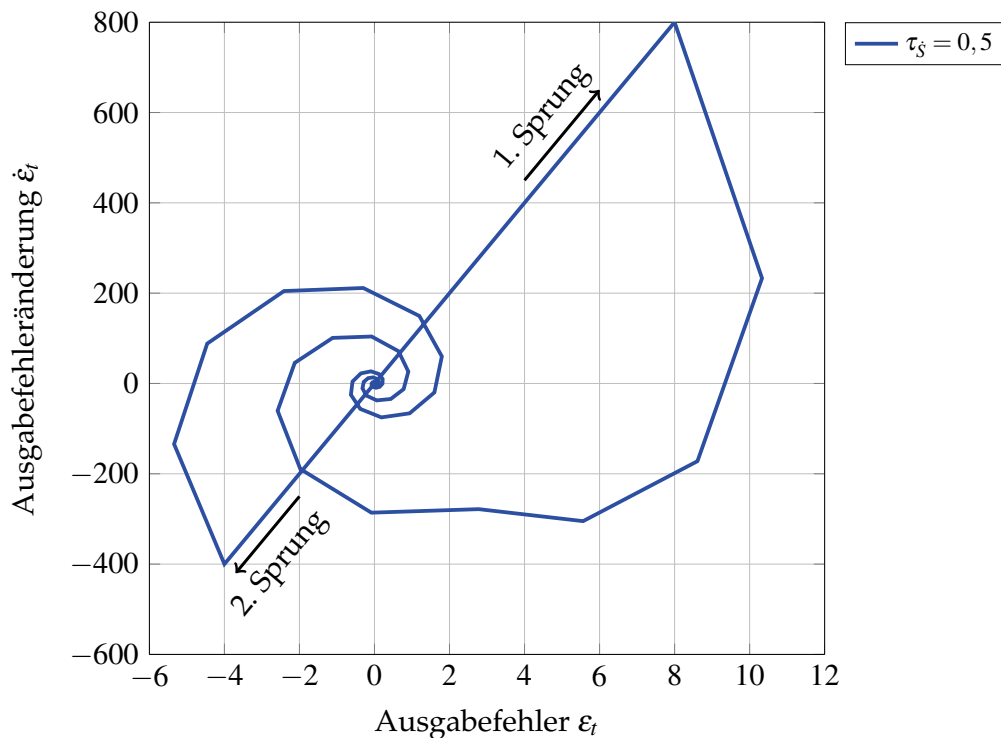


Abbildung 4.23: Phasenportrait eines mit dem Gleitzustandsverfahren zweiter Ordnung trainierten neuronalen Netzwerks mit zwei Sprüngen der Zielgröße

ist für den Zeitschritt direkt nach dem ersten Sprung das schon vom Gleitzustandstraining erster Ordnung bekannte Ansteigen des Fehlers ohne σ^* -Korrektur bei kleineren λ zu erkennen. Im Anschluss daran führt das Gleitzustandstraining zweiter Ordnung den Ausgabefehler ϵ_t stabil in den Ursprung. Generell kann die Anwendbarkeit des Twisting-Algorithmus auf das Training neuronaler Netzwerke somit bestätigt werden. Durch die Erweiterung der Lernrate um den Anteil μ_s ändert sich somit die Zustandstrajektorie von einem strikt durch S_t vorgegebene Verlauf aus Abbildung 4.8 zu einer wirbelartigen Trajektorie als Reaktion auf das zusätzliche Schalten an \dot{S}_t . Das dabei hervorgerufene schwingende Verhalten um $\epsilon_t = \dot{\epsilon}_t = 0$ wie es in Abbildung 4.23 zu sehen ist, stellt in dieser Form zunächst keine Verbesserung des Trainingsprozesses in Bezug auf die Konvergenzgeschwindigkeit dar. Aus diesem Grund gilt es für die weitere Untersuchung, die Abhängigkeit des Trainingserfolgs von den Intervallparametern τ und τ_s sowie der Lernratenbegrenzung durch das Stabilitätsintervall des Verfahrens erster Ordnung zu betrachten.

In Abbildung 4.24 sind die Zustandstrajektorien von unterschiedlichen Trainingsläufen mit einem Wertebereich von $\tau_s = [0,2 : 0,7]$ und konstantem $\tau = 0,5$ dargestellt. Aus Gründen der Übersichtlichkeit ist dabei nur der erste Sprung der Führungsgröße abgebildet, um eine bessere Vergleichbarkeit der wirbelnden Trajektorien zu ermöglichen.

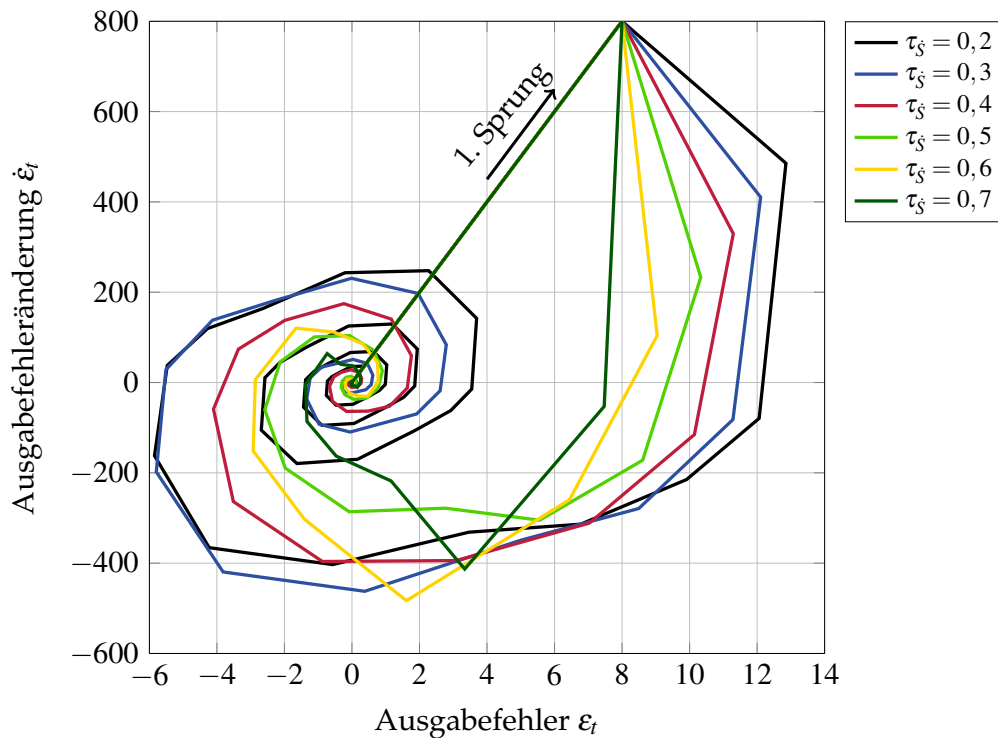


Abbildung 4.24: Phasenportrait eines mit dem Gleitzustandsverfahren zweiter Ordnung trainierten neuronalen Netzwerks mit einem Sprung der Zielgröße

Wie schon für das Verfahren erster Ordnung ist eine eindeutige Abhängigkeit der Fehlerzustände von der Wahl des Intervallfaktors zu erkennen. So führt eine Erhöhung von τ_s zu einer schnelleren Annäherung an den Ursprung, wohingegen eine Reduktion des Parameters ein vermehrtes Schwingen zur Folge hat. Darüber hinaus ist zu sehen, dass für $\tau_s = 0,7$ der Ausgabefehler bereits ohne σ^* -Korrektur im ersten Zeitschritt abnimmt, wohingegen der Fehler für alle weiteren Intervallfaktoren zunächst zunimmt. Für die detailliertere Untersuchung des Konvergenzverhaltens sind in Abbildung 4.25 die zeitlichen Verläufe der Ausgabegrößen des Netzwerks sowie der Schaltfunktionswerte S_t für den ersten Sprung dargestellt. Dabei wird sich analog der Untersuchung des Ansatzes erster Ordnung auf drei exemplarische Werte von τ_s beschränkt. Es ist gut die bereits im Phasenportrait zu sehende Reduktion der Schwingung um die Zielgröße bei steigendem Intervallparameter zu erkennen. So nimmt sowohl die Amplitude als auch die Häufigkeit der Überschwinger durch die Erhöhung der \dot{S}_t -abhängigen Lernrate μ_s ab. Entsprechende Überschwinger und deren Reduktion durch Erhöhung von τ_s finden sich auch im dargestellten Verlauf der Schaltfunktionswerte. Dabei ist, im Vergleich zu den Untersuchungen des Verfahren erster Ordnung aus Abbildung 4.9, ein deutlich unruhiger Verlauf zu sehen. So kommt es trotz

einer Wahl von $\tau = 0,5$ für den Intervallparameter der μ -Berechnung zu einer mehrfachen Überschreitung der Schaltlinie bei $S = 0$ für alle Werte von $\tau_{\dot{S}}$.

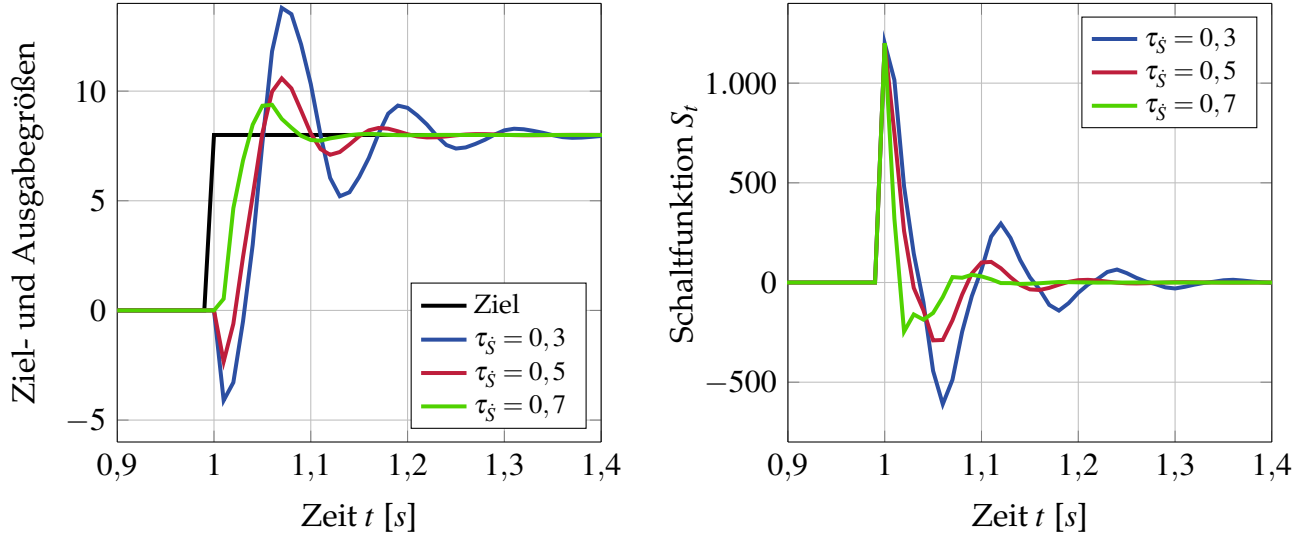


Abbildung 4.25: Netzausgaben und Schaltfunktionswerte für verschiedene Intervallparameter bei einem Sprung der Zielgröße

Ebenfalls der rechten Abbildung zu entnehmen, ist der nicht monoton abnehmende Schaltfunktionsbetrag für $\tau_{\dot{S}} = 0,7$. So ist zu erkennen, dass der Betrag stellenweise zunächst zunimmt, bevor er im folgenden Zeitschritt abgebaut wird. In diesen seltenen Fällen wird die Konvergenzbedingung $|S_{t+1,i}| \leq |S_{t,i}|$ nicht eingehalten, was es im Hinblick auf die Stabilität des Trainings auszuschließen gilt. Zu diesem Zweck wird für die folgenden Untersuchungen die kombinierte Lernrate μ_{Tw} mit den Intervallgrenzen aus Gleichung 4.28 limitiert. Die auf Grundlage der betragsmäßigen Schaltfunktionsabnahme in jedem Zeitschritt berechneten Grenzen stellen die Erreichbarkeit des Gleitzustands in jedem Zeitschritt sicher. Dabei ist es zweckmäßig, nicht exakt die obere und untere Lernratenbegrenzungen mit

$$\bar{\mu} = \frac{a_i + S_{t,i}}{b_i} \quad \text{und} \quad \underline{\mu} = \frac{a_i - S_{t,i}}{b_i} \quad (4.64)$$

zur Limitierung der kombinierten Lernrate zu verwenden, sondern die in Abschnitt 4.3.3 untersuchte Stabilitätsreserve von 20% zu berücksichtigen. Zu diesem Zweck wird der Index SR für die verkleinerten Intervallgrenzen eingeführt. Für die Begrenzung der kombinierte Lernrate μ_{Tw} , zur Garantie einer monoton abnehmenden Schaltfunktion, kann somit die folgende Korrektur vorgenommen werden:

$$\mu_{Tw} = \begin{cases} \bar{\mu}_{SR} & \text{für } \mu + \mu_{\dot{S}} > \bar{\mu}_{SR} \\ \mu + \mu_{\dot{S}} & \text{für } \underline{\mu}_{SR} < \mu + \mu_{\dot{S}} < \bar{\mu}_{SR} \\ \underline{\mu}_{SR} & \text{für } \mu + \mu_{\dot{S}} < \underline{\mu}_{SR} \end{cases} \quad (4.65)$$

Auf diese Weise wird die berechnete Lernrate des Gleitzustandstrainings zweiter Ordnung zu jeder Zeit innerhalb der stabilen Grenzen des Verfahrens erster Ordnung gehalten. Im

Falle einer zunächst berechneten Überschreitung dieser Grenzen durch μ_{Tw} fällt der Ansatz somit auf das Verhalten des Gleitzustandstrainings erster Ordnung zurück. Deutlich wird dies beispielsweise in dem in Abbildung 4.26 dargestellten Phasenportrait des ersten Führungsgrößensprungs.

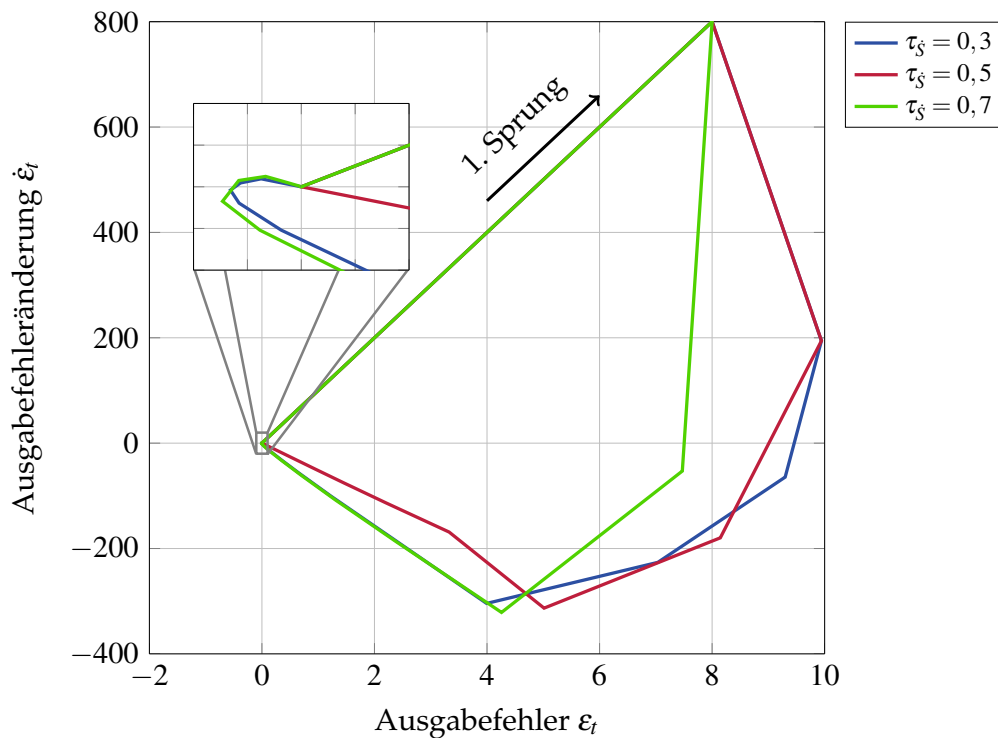


Abbildung 4.26: Phasenportrait eines mit dem Gleitzustandsverfahren zweiter Ordnung trainierten neuronalen Netzwerks mit Begrenzung der kombinierten Lernrate durch das Verfahrens erster Ordnung

Im Gegensatz zu Abbildung 4.24 sind keine ausgeprägten wirbelartigen Trajektorien um die Ruhelage mehr zu erkennen. Folglich führt die Begrenzung der Lernrate zu einer deutlichen Reduktion der Schwingungen. Ausschließlich kurz vor der Erreichung des Ursprungs sind die vom Twisting-Algorithmus bekannten abgerundeten Zustandstrajektorien, die einem einmaligen Überschießen der Zielgröße entsprechen, zu erkennen.

In Abbildung 4.27 sind die dazugehörigen Zeitverläufe der Netzausgaben sowie der Schaltsfunktionswerte dargestellt. Im linken Teil der Abbildung ist gut der schon im Phasenportrait zu sehende Rückgang der Schwingungen um die Zielgröße zu erkennen. Lediglich kurz vor dem Erreichen des Endwerts kommt es für $\tau_{\dot{s}} = 0,3$ und $\tau_{\dot{s}} = 0,7$ zu einem einzelnen Überschwinger der Ausgabegröße. Im rechten Teil der Abbildung ist der nun ausschließlich monoton-fallende Verlauf des Schaltsfunktionsbetrags nach dem Sprung zu sehen. Im direkten Vergleich zu Abbildung 4.25 kommt es nach dem Sprung zu keinem betragsmäßigen Anstieg von S_t mehr.

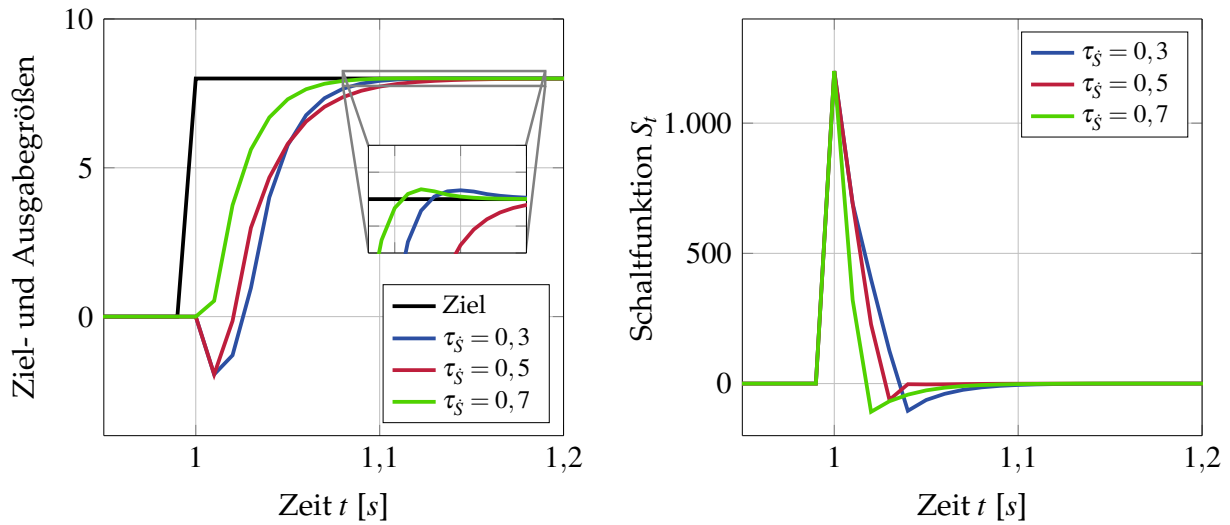


Abbildung 4.27: Netzausgaben und Schaltfunktionswerte mit Begrenzung der kombinierten Lernrate durch das Verfahrens erster Ordnung

Somit wird durch die Verwendung der Lernratenbegrenzung aus Gleichung 4.65 die Erreichbarkeitsbedingung $|S_{t+1,i}| \leq |S_{t,i}|$ auch für das Verfahren zweiter Ordnung zu jedem Zeitpunkt erfüllt. Die letztendlich berechneten kombinierten Lernraten für die drei exemplarischen Werte von τ_{S_t} während des ersten Sprungs der Führungsgröße sind in Abbildung 4.28 dargestellt.

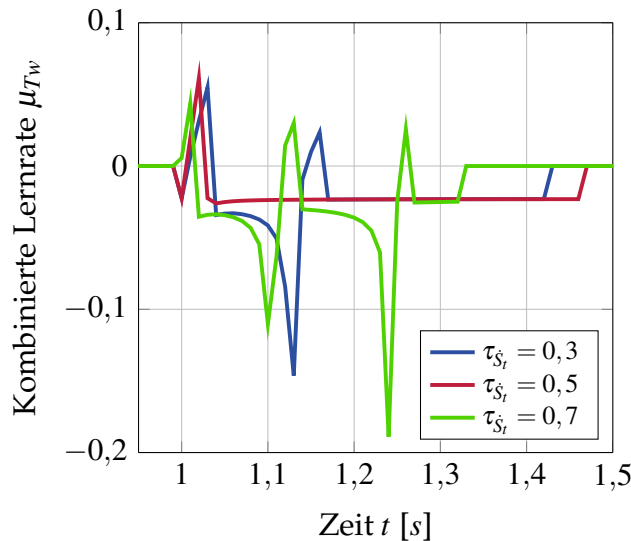


Abbildung 4.28: Kombinierte Lernrate für verschiedene Intervallparameter bei einem Sprung der Zielgröße

Auffällig dabei ist der nahezu konstante und an das Ergebnis des Verfahrens erster Ordnung erinnernde Verlauf für $\tau_{S_t} = 0,5$, wohingegen es für sowohl $\tau_{S_t} = 0,3$ als auch $\tau_{S_t} = 0,7$ zu größeren Sprüngen der Lernrate kommt. Dieses Ergebnis ist auf die schnellere Annäherung an $S_t = \dot{S}_t = 0$ für die Wahl von τ und τ_{S_t} in der Mitte der beiden aufgestellten Lernraten-

intervalle zurückzuführen. Auf diese Weise nähern sich auch die Grenzen von μ und $\mu_{\hat{S}_t}$ schneller der Intervallmitte bei $\frac{a}{b}$ an, was zu dem deutlich ruhigeren Lernratenverlauf führt.

Ein weiteres den Simulationen zu entnehmendes Ergebnis ist die in der Theorie erwähnte Chattering-Reduktion des Twisting-Algorithmus. Sowohl in den Lernratenverläufen in Abbildung 4.28 als auch in den Verläufen eines Trainings mit $\tau = 0,7$ in Anhang A.4 kann dieser Effekt festgestellt werden. Wo die Wahl von $\tau > 0,5$ für das Verfahren erster Ordnung zu Chattering in der Lernrate führt, zeigt das Gleitzustandstraining zweiter Ordnung keine Anzeichen dieses schnellen Vorzeichenwechsels. So kann festgehalten werden, dass sowohl die charakteristische Zustandstrajektorie als auch die Chattering-Reduktion der Gleitzustandsregelung zweiter Ordnung auf das Training neuronaler Netzwerke übertragen werden kann.

4.4.3 Untersuchung der gegenseitigen Beeinflussung von τ , $\tau_{\hat{S}}$ und λ

Nachdem nun die Abhängigkeit des Netzwerktrainings zweiter Ordnung von $\tau_{\hat{S}}$ bei einem konstanten $\tau = 0,5$ bekannt ist, wird im Folgenden der Einfluss variierender τ und λ auf die zuvor untersuchte Abhängigkeit von $\tau_{\hat{S}}$ analysiert. Zu diesem Zweck wird im Folgenden das komplexe Trainingsszenario aus Abbildung 4.12 verwendet. Insbesondere in Hinblick auf das oftmals durch den Twisting-Algorithmus induzierte schwingende Verhalten gilt es, die Effekte der beiden Intervallparameter aufeinander zu untersuchen. Mit dem Ziel dem in Abbildung 4.27 zu sehenden Anstieg des Netzwerkfehlers im ersten Zeitschritt nach dem Führungsgrößensprung für kleine λ Rechnung zu tragen, wird in den folgenden Untersuchungen erneut die σ^* -Korrektur verwendet. Auf diese Weise wird sichergestellt, dass das Vorzeichen der Lernrate immer dem der Multiplikation von Schaltfunktion und Netzwerkfehler entspricht. Wie oben dargestellt, basiert dieses Vorgehen auf der Bedingung, dass $|\varepsilon_{t+1}| < |\varepsilon_t|$ gelten soll. Im Falle von $\text{sign}(\mu_{Tw}^*) \neq \text{sign}(S_{t,i} \cdot \varepsilon_{t,i})$ wird die Lernratenbestimmung somit zu $\mu_{Tw} = \frac{a_i}{b_i}$ vereinfacht. Nach der Definition von a_i und b_i aus den Gleichungen 4.15 und 4.16 wird auf diese Weise die Bedingung der Vorzeichengleichheit $\text{sign}(\mu_{Tw}^*) = \text{sign}(S_{t,i} \cdot \varepsilon_{t,i})$ zu jeder Zeit erfüllt.

Für die Analyse des komplexen Führungssignals wird ein Wertebereich von $\tau = [0, 2 \dots 0, 8]$ und $\tau_{\hat{S}} = [0, 2 \dots 0, 8]$ simuliert und der Mean-Square-Error des über 20 Sekunden erzeugten Netzwerkfehlers ε_t verglichen. Dabei wird jede Kombination der Intervallparameter für einen Wertebereich des Schaltfunktionsparameters von $\lambda = [50 \dots 950]$ untersucht. Der Übersichtlichkeit halber sind in Abbildung 4.29 nur die Ergebnisse für $\lambda = [50 \dots 550]$ dargestellt. Die Auswertungen des Fehlers für $\lambda = [650 \dots 950]$ sind in Abbildung A.5 in Anhang A zu sehen. Jeder der unterschiedlich farbigen Verläufe gibt das Trainingsergebnis für einen Wert von $\tau_{\hat{S}}$ an. So ist für jeweils ein gegebenes λ der Einfluss von sowohl τ als auch $\tau_{\hat{S}}$ abzulesen. Zusätzlich ist als gestrichelte Linie das Ergebnis des Gleitzustandstrainings erster Ordnung SMC-FO für das gegebene λ zu sehen. Somit stellt der grau eingefärbte Bereich den Parameterraum dar, indem die gewählte Kombination zu einer Erhöhung des MSE für das jeweils

gewählte τ führt.

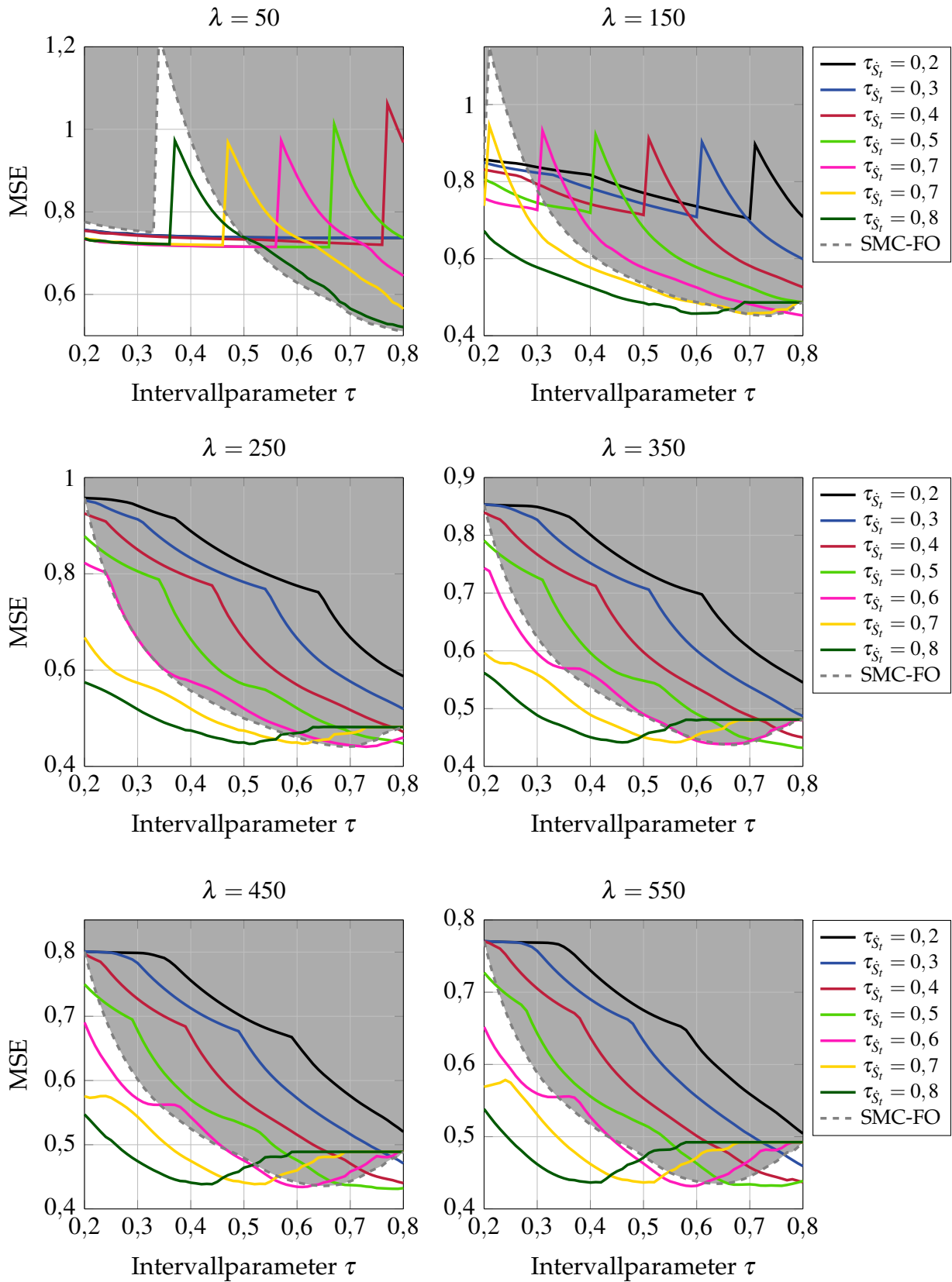


Abbildung 4.29: MSE Vergleich in Abhängigkeit von τ , $\tau_{\dot{S}}$ und λ für $\lambda = [50...550]$

Für $\lambda = 50$ sowie $\lambda = 150$ ist deutlich der Einfluss der σ^* -Korrektur zu erkennen. So kommt es zu einem zunächst sprunghaften Anstieg des Fehlers im Moment der nicht länger notwendigen Korrektur auf $\mu_{Tw} = \frac{a}{b}$, bevor der Fehler im Anschluss wieder abnimmt. Für die weiteren Werte von λ ist ein solches Verhalten nicht zu sehen, da die Bedingung $|\varepsilon_{t+1}| < |\varepsilon_t|$ bereits ohne σ^* -Korrektur erfüllt wird. Ebenfalls gut zu sehen sind die für steigende λ immer weiter in Richtung kleiner τ wandernde MSE-Minima. In Richtung steigender τ wird eine Grenze erreicht, bei der die kombinierte Lernrate μ_{Tw} die Stabilitätsgrenzen des Verfahrens erster Ordnung verlassen würde, so dass im Anschluss für alle Werte von $\tau_{\hat{g}}$ und τ durch die Begrenzung der Lernrate der gleiche MSE erzeugt wird. Dieser entspricht jeweils dem Wert des Verfahrens erster Ordnung bei $\tau = 0,8$, da dies der Grenze zur Wahrung der aufgestellten Stabilitätsreserve entspricht.

Generell kann für alle untersuchten λ festgestellt werden, dass jeweils ein minimaler Fehler existiert, dem sich verschiedene Parameterkombinationen annähern, jedoch kein klar zu identifizierendes Optimum von τ und $\tau_{\hat{g}}$ bestimmt werden kann. Weiterhin geht aus den Untersuchungen hervor, dass der Einfluss der Intervallparameter τ und $\tau_{\hat{g}}$ vom Trainings-szenario abhängt. Aus diesem Grund ist neben der Quantifizierung des Netzwerkfehlers für diese nur exemplarischen Trainingsbeispiele besonders die Dynamik der Netzwerkausgabe bei der Wahl unterschiedlicher Parameterkombinationen von Relevanz. Für die Untersuchung des Einflusses von τ und $\tau_{\hat{g}}$ auf die Dynamik werden im folgenden Simulationen für Parameterkombinationen betrachtet, die zu einem identischen MSE führen. Als Vergleich dient dabei das in Tabelle 4.5 gegebene Gleitzustandstraining erster Ordnung SMC-FO1. In Abbildung 4.30 sind dafür die Fehler-Verläufe mit zusätzlicher Markierung des MSE-Werts für das Vergleichsverfahren dargestellt. Zur Identifizierung gleicher MSE-Bereiche ist eine horizontale Hilfslinie über den gesamten Wertebereich von τ dargestellt.

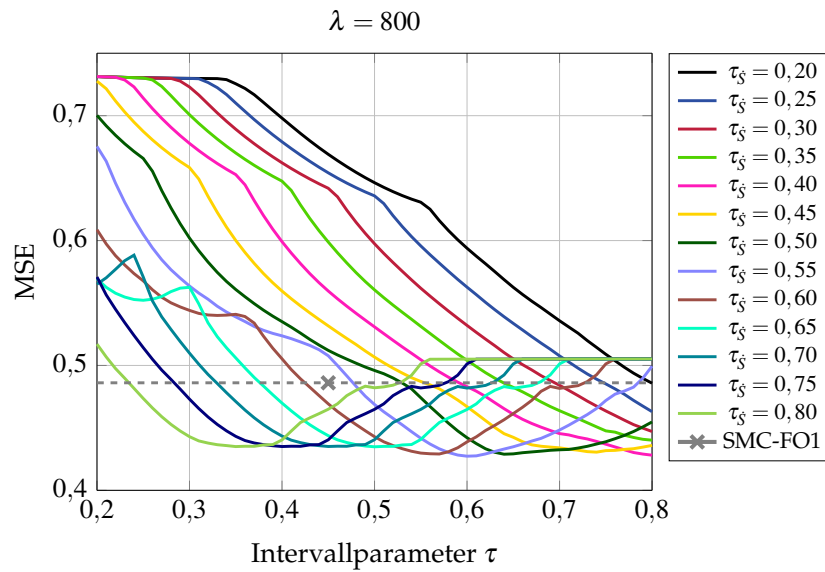


Abbildung 4.30: MSE Vergleich in Abhängigkeit von τ und τ_S für $\lambda = 800$ mit Referenzwert des Verfahrens erster Ordnung

Für die gefundenen Wertepaare von τ und τ_S mit gleichem MSE können nun die Netzausgaben verglichen werden. Die ausgewählten Kombinationen der beiden Intervallparameter sind in Tabelle 4.7 aufgeführt.

τ	0,43	0,33	0,24	0,71	0,61	0,51
τ_S	0,60	0,70	0,80	0,60	0,70	0,80

Tabelle 4.7: Parameterkombinationen zur Erreichung des MSE von SMC-FO1

Dem Vorgehen für die Verfahren erster Ordnung folgend, sind in Abbildung 4.31 die Netzausgaben der unterschiedlich parametrisierten Verfahren zweiter Ordnung für einen Ausschnitt der komplexen Führungsgrößenvorgabe dargestellt.

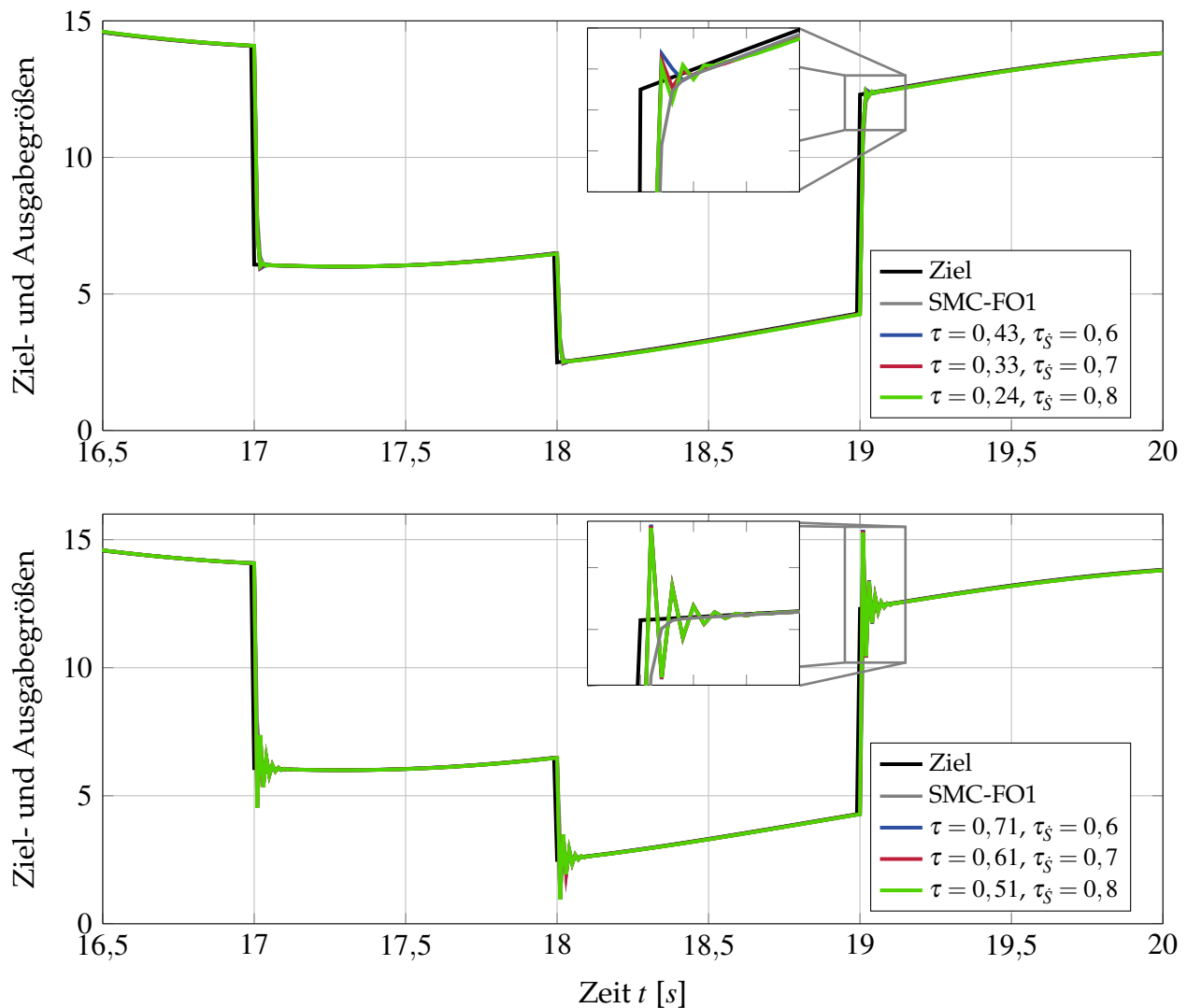


Abbildung 4.31: Vergleich der Netzwerkausgaben für drei verschiedene Kombinationen von τ und τ_S mit SMC-FO1

Hierbei ist zum Vergleich jeweils der Verlauf des Verfahrens erster Ordnung SMC-FO1 mit abgebildet. Die obere Hälfte der Darstellung zeigt das Ergebnis der Parameterkombinationen, die in Abbildung 4.30 links des Fehlerminimums liegen. Entsprechend sind in der unteren Hälfte die Verläufe für Werte mit einem τ rechts vom jeweiligen Minimum zu sehen. Wichtig zu beachten ist, dass alle abgebildeten Verläufe, inklusive denen von SMC-FO1, einen nahezu identischen Mean-Square-Error aufweisen.

Wie zu erwarten, ist für alle drei gezeigten Wertepaare mit einem τ rechts der Minima deutliches Oszillieren um die Zielgröße bei allen abgebildeten Sprüngen zu erkennen. Im Vergleich dazu zeigt die Dynamik der Verläufe in der oberen Hälfte der Abbildung ein deutlich ruhigeres Verhalten. Darüber hinaus sind die gezeigten Ergebnisse der jeweiligen drei Parameterkombinationen nicht nur im MSE sondern auch in ihren Ausgaben sehr ähnlich. Mit anderen Worten kann ein nahezu lineares Verhalten für die beiden Intervallpa-

parameter festgestellt werden, so dass die Verringerung von τ und die Vergrößerung von τ_s letzten Endes wieder zu einem nahezu identischen Ergebnis führt.

Zusammenfassend kann festgehalten werden, dass das Gleitzustandsverfahren zweiter Ordnung in Anlehnung an den Twisting-Algorithmus auf das Training neuronaler Netzwerke übertragen werden kann. Durch die Einführung der erweiterten dynamischen Lernratenberechnung von μ_{Tw} können in einigen Parameterbereichen relative Verbesserungen des Mean-Square-Errors für ein gegebenes τ erreicht werden. In der absoluten Betrachtung des gesamten Intervallbereichs ist es vor allem die Veränderung der Dynamik des Netzwerkausgangs, die durch den Einsatz des Verfahrens zweiter Ordnung erreicht werden kann. So ist es möglich, bei gleichbleibendem MSE eine, überschwingungsfreien Verlauf des Verfahrens erster Ordnung initial durch einen Überschwinger zu beschleunigen oder einen zu Schwingungen neigenden Verlauf durch die Hinzunahme von τ_s zu dämpfen. Bei der Wahl des geeigneten Verfahrens handelt es sich somit um eine ingenieurmäßige Fragestellung, welche Dynamik vom neuronalen Netzwerk während der jeweiligen Regelungsaufgabe bevorzugt wird.

Unabhängig von der schlussendlichen Parameterwahl ist durch die auf Erreichbarkeits- und Existenzbedingungen basierenden Berechnungen der Lernrate sichergestellt, dass diese innerhalb stabiler Grenzen bestimmt werden und zur Erreichung des robusten Gleitzustands führen. Auf diese Weise ist es nicht länger notwendig, eine konservative Auslegung der Lernrate vorzunehmen, um mögliche Instabilitäten bei zu groß gewählten Werten zu verhindern. Darüber hinaus kann durch die Kontraktionsbedingungen eine Aussage über das Verhalten der Netzwerkzustände im kommenden Zeitschritt getroffen werden, was die Transparenz des Netzwerktrainings gegenüber reinen auf Optimierung basierenden Verfahren wie dem Backpropagation-Algorithmus erhöht. Untersuchungen der unterschiedlichen Trainingsverfahren zur Unterstützung einer nichtlinearen Inversionsregelung sind in Kapitel 7 dargestellt.

Kapitel 5

Regelungskonzept

Die im vorherigen Kapitel vorgestellten Gleitzustandsverfahren werden in diesem Kapitel nun auf das Training neuronaler Netzwerke als Erweiterung einer nichtlinearen Regelungsarchitektur angewendet. Dabei handelt es sich um eine exemplarische Einsatzmöglichkeit von gleitzustandstrainierten Netzwerken, die jedoch durch ihre aktuelle Relevanz im Feld der adaptiven und fehlertoleranten Regelung automatischer Systeme anschaulich die Wichtigkeit hoher Konvergenzgeschwindigkeiten bei nachgewiesener Stabilität demonstriert.

Bei dem in dieser Arbeit verwendeten Regelungskonzept handelt es sich um die sogenannte nichtlineare dynamische Inversion. Die wesentliche Grundidee dieses modellbasierten Ansatzes ist es, einem ursprünglich nichtlinearem System ein arbeitspunktunabhängiges lineares Eingangs-/Ausgangsverhalten aufzuprägen, ohne dazu vereinfachende Annahmen einer klassischen Linearisierung treffen zu müssen [38, 39]. Umgesetzt wird dies durch eine sogenannte linearisierende Zustandsrückführung, durch die es möglich wird, bekannte und modellierbare Nichtlinearitäten des Systems im Eingangs-/Ausgangsverhalten zu negieren. Aus diesem Grund sind alternative Bezeichnungen des Verfahrens auch Eingangs-/Ausgangslinearisierung oder *feedback linearisation*. Die Regelgüte der dynamischen Inversion hängt dabei maßgeblich von der Qualität des vorhandenen Modellwissens ab. Folglich weist das Verfahren eine deutliche Anfälligkeit gegenüber Parameter- und Modellunsicherheiten auf. Durch ihre universelle Approximationsfähigkeit und der dadurch möglichen Kompensation vorhandener Modellfehler eignen sich insbesondere *online*-trainierte neuronale Netzwerke für die Erhöhung der Robustheit und Leistungsfähigkeit des Regelungskonzeptes [34, 46, 48, 90, 110]. Im weiteren Verlauf dieser Arbeit wird demonstriert, dass neuronale Netzwerke über die Kompensation von Modellunsicherheiten hinaus, auch zum Ausgleich von sprunghaft auftretenden Systemschäden und Steuerdegradationen geeignet sind.

Erste Forschungsarbeiten zu sowohl der Theorie als auch später der praktischen Umsetzung von Regelungsverfahren auf Basis dynamischer Inversion wurden ab Ende der 1980er Jahre und im Anschluss mehrheitlich in den 1990er Jahren auf dem Gebiet der Flugregelung durchgeführt [40, 44, 45, 111–115]. Dabei wurden neben zivilen und militärischen Flächen-

flugzeugen auch Drehflügler als Regelungsstrecke untersucht. Dem oben beschriebenen Problem von nur unvollständigem Modellwissen wurde ab Mitte der 1990er Jahre mit der Approximationsfähigkeit von neuronalen Netzwerken begegnet [32–35, 51]. Dabei hat insbesondere ein Forschungsteam am Georgia Institute of Technology in den USA die Arbeiten federführend vorangetrieben. Im Rahmen dieser Untersuchungen konnte der bis dahin bereits umfänglich untersuchte Inversionsregelkreis auch um das sogenannte Pseudo-Control Hedging (vgl. Abschnitt 5.1.5) erweitert werden [37, 116]. Dabei handelt es sich um ein Verfahren, bei dem unmodellerte nichtlineare Ausschlags- und Ratenbegrenzungen der Aktuatorik bereits in der Referenzgrößenerzeugung berücksichtigt werden. Auf diese Weise kann das unbegrenzte Anwachsen von Integratoren der Regelungsschleifen (*integrator wind-up*) und das Anlernen einer gesättigten Stelldynamik durch neuronale Netzwerke verhindert werden, ohne die mögliche Steuermomentenvorgabe künstlich konservativ auslegen zu müssen.

Neben der Anwendung für mannttragende Luftfahrzeuge hat sich gerade das Feld der unbemannten Flugsysteme als wachsender Forschungsschwerpunkt für adaptive und nichtlineare Regelung hervor getan. Durch die zumeist nur begrenzte Parameteridentifikation solch oftmals nur sehr kleinen Flugsystemen kommt es bei der Verwendung modellbasierter Regelungsverfahren zu erheblichen Parameter- und Modellunsicherheiten. Darüber hinaus weisen solch meist leichten Systeme eine hohe Anfälligkeit gegenüber starkem Wind und Turbulenz auf, so dass es bei klassisch an einzelnen Arbeitspunkten ausgelegte Linearreglern bei turbulenten Atmosphärenbedingungen oftmals zu erheblichen Verlusten in der Regelgüte kommt. Im Vergleich dazu konnten um neuronale Netzwerke erweiterte Inversionsregler eine Reduktion des Einflusses von Wind und ungenauen Modellwissen auf das Regelungsergebnis bewirken und auf diese Weise sowohl in der Simulation als auch im Flugversuch eine gesteigerte Leistungsfähigkeit demonstrieren [10, 46–50].

Ein weiterer Vorteil des Einsatzes von neuronalen Netzen und des im Folgenden vorgestellten Inversionsreglers ist deren Eignung für die Erweiterung um Fehlerdetektions- und Rekonfigurationskonzepte. Diese erlauben es unter anderem kritische Ausfallsituationen stabil zu kompensieren, die bei der Verwendung klassischer Regelungsarchitekturen zu einem möglichen Absturz geführt hätten [19, 46, 117–120]. Die Summe der aufgelisteten Eigenschaften haben über die letzten Jahre dabei die Bedeutung der in Abschnitt 2.3.2 angesprochenen Arbeiten an Zulassungs- und Verifikationsmethoden von adaptiven Regelungssystemen weiter verstärkt. Insbesondere durch einen möglichen Transfer der adaptiven Regelungskonzepte auf das Forschungsgebiet des automatischen Fahrens rücken die Methoden zur Bewertung und Zertifizierung von adaptiven Strukturen aktuell noch weiter in den Fokus wissenschaftlicher und industrieller Arbeiten [52, 53, 69, 121, 122].

5.1 Nichtlineare dynamische Inversion

In diesem Abschnitt werden die theoretischen Grundlagen für den innerhalb dieser Arbeit verwendeten dynamischen Inversionsregler dargestellt. Dabei orientiert sich sowohl die Struktur als auch die dargestellte Theorie maßgeblich an den Ausarbeitungen in [46,48]. Vertiefende Darstellungen des theoretischen Konzepts sind für die allgemeine Betrachtung von nichtlinearen Systemen in [38,39] zu finden.

5.1.1 Grundlegendes Prinzip der dynamischen Inversion

Für die dargestellten Grundlagen werden nichtlineare dynamische Mehrgrößensysteme (engl. *multiple input, multiple output* - MIMO) mit gleicher Anzahl von Eingangs- und Ausgangsgrößen sowie Linearität im Steuerverhalten betrachtet:

$$\begin{aligned}\dot{\vec{x}} &= \vec{f}(\vec{x}) + \mathbf{G}(\vec{x}) \cdot \vec{u}, \\ \vec{y} &= \vec{h}(\vec{x}).\end{aligned}\tag{5.1}$$

Dabei handelt es sich bei $\vec{x}_{n \times 1} = [x_1, \dots, x_n]^T$ um den Zustandsvektor der Länge n sowie bei $\vec{u}_{m \times 1} = [u_1, \dots, u_m]^T$ und $\vec{y}_{m \times 1} = [y_1, \dots, y_m]^T$ um den Ein- bzw. Ausgangsvektor des Systems mit der Länge m . Bei allen Abbildungen $\vec{f}(\vec{x})$, $\mathbf{G}(\vec{x})$ und $\vec{h}(\vec{x})$ wird davon ausgegangen, dass sie ausreichend glatt und somit mehrfach stetig differenzierbar sind. Zusammengesetzt ist die Matrix $\mathbf{G}_{n \times m}(\vec{x})$ dabei aus m Vektorfeldern $\vec{g}_{n \times 1}(\vec{x})$, welche, wie auch die Vektorfelder $\vec{f}_{n \times 1}(\vec{x})$ und $\vec{h}_{m \times 1}(\vec{x})$, aus nichtlinearen Funktionen des Zustandsvektors \vec{x} bestehen.

Ein solches System kann nun durch eine nichtlineare Zustandsrückführung in Form einer geeigneten Steuergröße \vec{u} in ein äquivalentes Linearsystem transformiert werden. Dabei bietet die Theorie der *feedback linearization* sowohl die Möglichkeit der Eingangs-/Zustandslinearisierung sowie der Eingangs-/Ausgangslinearisierung, wie sie in dieser Arbeit verwendet wird. Anhand eines einfachen Beispiels aus [39] soll dafür zunächst die Notwendigkeit einer geeigneten Systembeschreibung für die Erzeugung eines linearen Eingangs-/Ausgangsverhaltens demonstriert werden.

Beispiel: Feedback Linearization

Es soll das einfache System

$$\begin{aligned}\dot{x}_1 &= a \cdot \sin(x_2) \\ \dot{x}_2 &= -x_1^2 + u \\ y &= x_2\end{aligned}\tag{5.2}$$

mit Nichtlinearität in der Zustandsgröße \dot{x}_1 betrachtet werden. Es ist direkt zu erkennen, dass im Falle einer angestrebten Eingangs-/Zustandslinearisierung durch die Steuergröße u , die Nichtlinearität in Form von $\sin(x_2)$ nicht direkt ausgeglichen werden kann, da keine

direkte Beeinflussung der beiden Größen voneinander vorliegt. Aus diesem Grund ist es notwendig, eine Zustandstransformation vorzunehmen, um einen direkten Zusammenhang zwischen der Steuergröße und der auszugleichenden Nichtlinearität herzustellen. Mit der Transformation $z = \Phi(x)$, umgesetzt durch

$$z_1 = x_1 \quad \text{und} \quad z_2 = a \cdot \sin(x_2) \quad (5.3)$$

ergibt sich

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= a \cdot \cos(x_2) \cdot (-x_1^2 + u). \end{aligned} \quad (5.4)$$

Hier kann nun in der zweiten Gleichung ein direkter Zusammenhang zwischen der Steuergröße und der Nichtlinearität gefunden werden. Mit der geeigneten Wahl der zustandsrückführenden Steuergröße u zu

$$u = x_1^2 + \frac{1}{a \cdot \cos(x_2)} \cdot v \quad (5.5)$$

kann der Einfluss des nichtlinearen Terms $\cos(x_2)$ ausgeglichen werden, so dass ein eingangs-/zustandslinearisiertes Systems der Form

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= v \\ y &= \sin^{-1} \left(\frac{z_2}{a} \right) \end{aligned} \quad (5.6)$$

übrig bleibt. Bei v handelt es sich um die Steuergröße des transformierten Systems, die folglich als Pseudosteuergröße bezeichnet wird. Das ursprünglich nichtlineare Zustandsverhalten konnte somit durch eine Zustandstransformation und eine linearisierende Zustandsrückführung kompensiert und in ein lineares Verhalten der transformierten Systemzustände umgeformt werden. Die vorgestellte Linearisierung betrifft dabei nur das Eingangs-/Zustandsverhalten, so dass die Ausgangsgröße weiterhin nichtlinear beeinflusst bleibt.

Ist es nun das Ziel, das Eingangs-/Ausgangsverhalten des Systems durch die Zustandsrückführung zu linearisieren, kann diese zu

$$u = x_1^2 + v \quad (5.7)$$

gewählt werden. Auf diese Weise ergibt sich eine lineare Eingangs-/Ausgangsabbildung, wie es in Gleichung 5.8 zu sehen ist:

$$\begin{aligned} \dot{x}_2 &= v \\ y &= x_2. \end{aligned} \quad (5.8)$$

Es ist sofort ersichtlich, dass der Ausgang y nicht länger von einer Nichtlinearität beeinflusst wird, so dass die Zustandsrückführung zu der gewünschten Linearität zwischen Eingangs- und Ausgangsgröße geführt hat. Die Pseudosteuergröße gibt in diesem Beispiel die erste Ableitung der gewünschten Ausgabegröße vor, so dass diese abschließend einmal integriert werden muss, um die finale und linear von v abhängende Systemantwort zu erhalten.

Der relative Grad

Aus dem vorangegangenen Beispiel wird deutlich, dass für die Linearisierung des Eingangs-/Ausgangsverhaltens neben der benötigten Zustandsrückführung eine bestimmte Form des Systems gegeben sein muss, die einen Zusammenhang zwischen der Eingangsgröße \vec{u} und der Ausgangsgröße \vec{y} aufweist. Liegt diese Form nicht vor, muss das System entsprechend transformiert werden. Wichtig dafür ist die Existenz eines relativen Grades, wie er schon in Abschnitt 3.2 für die Definition von Gleitordnungen eingeführt wurde [38,39]. Zur Veranschaulichung dieser wichtigen Größe als Bedingung für die dynamische Inversion soll der Übersichtlichkeit halber zunächst ein Eingrößensystem (engl. *single input, single output - SISO*) nach Gleichung 5.9 betrachtet werden.

$$\begin{aligned}\dot{\vec{x}} &= \vec{f}(\vec{x}) + \vec{g}(\vec{x}) \cdot u, \\ y &= h(\vec{x}).\end{aligned}\tag{5.9}$$

Wie oben beschrieben, wird für die angestrebte Linearisierung zunächst der Einfluss der Eingangsgröße auf den Systemausgang benötigt. Ist dieser in der Ausgangsgleichung nicht gegeben, kann diese solange abgeleitet werden, bis ein Zusammenhang zwischen der r -ten Ableitung $y^{(r)}$ und der Steuergröße u gefunden wurde. Die Anzahl der benötigten Ableitungen gibt dabei den relativen Grad r an. Leitet man die Ausgangsgröße ab, ergibt sich:

$$\dot{y} = \frac{\partial y}{\partial x} \cdot \frac{\partial x}{\partial t} = \frac{\partial h}{\partial \vec{x}} \cdot \dot{\vec{x}} = \frac{\partial h}{\partial \vec{x}} \cdot [\vec{f}(\vec{x}) + \vec{g}(\vec{x}) \cdot u] = L_f \cdot h(\vec{x}) + L_g \cdot h(\vec{x}) \cdot u.\tag{5.10}$$

Dabei handelt es sich bei $L_f \cdot h(\vec{x})$ und $L_g \cdot h(\vec{x})$ um sogenannte Lie-Ableitungen, die der Ableitung einer Funktion entlang eines Vektorfeldes entsprechen [38,39]. Ein mehrfaches Ableiten eines Systems mittels Lie-Ableitung ist wie in Gleichung 5.11 gezeigt, ebenfalls sehr einfach umsetzbar:

$$L_f^k \cdot h(\vec{x}) = L_f \left(L_f^{k-1} \cdot h(\vec{x}) \right).\tag{5.11}$$

Kommt es also dazu, dass die Eingangsgröße u an einem untersuchten Punkt $x = x_0$ keinen Einfluss auf die erste Ableitung von y hat, so dass folglich $L_g \cdot h(\vec{x}_0) = 0$ gilt, vereinfacht sich Ausdruck 5.10 zu:

$$\dot{y} = L_f \cdot h(\vec{x}_0).\tag{5.12}$$

Entsprechend dem oben beschriebenen Vorgehen wird im Anschluss die Ausgangsgleichung so lange weiter abgeleitet, bis in der r -ten Zeitableitung die Steuergröße u auftaucht:

$$y^{(r)} = L_f^r \cdot h(\vec{x}) + L_g \cdot L_f^{r-1} \cdot h(\vec{x}) \cdot u\tag{5.13}$$

Der relative Grad gibt somit die Ableitung der Ausgangsgröße y an, die sprungfähig auf Änderungen der Steuergröße reagiert. Im Umkehrschluss gibt der relative Grad die durch r Integrationen entstehende Verzögerung des Systems an, bis eine Steuergrößenänderung einen Einfluss auf die eigentliche Ausgangsgröße besitzt. Der dargestellte Ansatz zur Berechnung des relativen Grades lässt sich analog auf ein Mehrgrößensystem nach Gleichung 5.1 übertragen. Für den Vektor der abgeleiteten Ausgangsgleichungen folgt somit:

$$\begin{bmatrix} y_1^{r_1} \\ y_2^{r_2} \\ \vdots \\ y_m^{r_m} \end{bmatrix} = \begin{bmatrix} L_f^{r_1} \cdot h_1(\vec{x}) \\ L_f^{r_2} \cdot h_2(\vec{x}) \\ \vdots \\ L_f^{r_m} \cdot h_m(\vec{x}) \end{bmatrix} + \begin{bmatrix} L_{g_1} \cdot L_f^{r_1-1} \cdot h_1(\vec{x}) & \cdots & L_{g_m} \cdot L_f^{r_1-1} \cdot h_1(\vec{x}) \\ L_{g_1} \cdot L_f^{r_2-1} \cdot h_2(\vec{x}) & \cdots & L_{g_m} \cdot L_f^{r_2-1} \cdot h_2(\vec{x}) \\ \vdots & \ddots & \vdots \\ L_{g_1} \cdot L_f^{r_m-1} \cdot h_m(\vec{x}) & \cdots & L_{g_m} \cdot L_f^{r_m-1} \cdot h_m(\vec{x}) \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}. \quad (5.14)$$

Dabei lässt sich der Ausdruck durch die Einführung des Vektors $\vec{b}(\vec{x})$ und der sogenannten Entkopplungsmatrix $\mathbf{A}(\vec{x})$ wie folgt vereinfachen:

$$\begin{bmatrix} y_i^{(r_i)} \end{bmatrix}_{m \times 1} = \vec{b}(\vec{x}) + \mathbf{A}(\vec{x}) \cdot \vec{u}. \quad (5.15)$$

Wie in Gleichung 5.14 zu sehen ist, existiert im Mehrgrößenfall nicht ein einzelner relativer Grad r sondern ein vektorieller relativer Grad des Systems mit der Ordnung n , der sich wie folgt zusammensetzt:

$$r = r_1 + \dots + r_m = \sum_{i=1}^m r_i \leq n. \quad (5.16)$$

Mit diesem Wissen lässt sich abschließend eine mathematische Formulierung für den (vektoriellen) relativen Grad eines Mehrgrößensystems nach [38, 39, 46] wie folgt definieren.

(Vektorieller) Relativer Grad eines Mehrgrößensystems

Ein nichtlineares Mehrgrößensystem der Form 5.1 besitzt an einem Punkt \vec{x}_0 einen (vektoriellen) relativen Grad r_1, \dots, r_m , wenn

- $L_{g_j} \cdot L_f^k \cdot h_i(\vec{x}) = 0 \quad \forall \quad 1 \leq i \leq m, \quad 1 \leq j \leq m, \quad k < r_i - 1 \quad \text{und} \quad \vec{x} \text{ in einer Umgebung von } \vec{x}_0$
- und die Entkopplungsmatrix $\mathbf{A}_{m \times m}(\vec{x})$ bei $\vec{x} = \vec{x}_0$ invertierbar ist. \diamond

Zustandstransformation

Unter der Voraussetzung, dass ein System einen wohldefinierten vektoriellen relativen Grad besitzt, kann nun die angekündigte Transformation $\vec{z} = \Phi(\vec{x})$ des Systems durchgeführt werden. Dabei ist es wie im obigen Beispiel gezeigt das Ziel, eine Systemformulierung aufzustellen, bei der die Nichtlinearitäten des Eingangs-/Ausgangsverhalten des Systems durch eine linearisierende Zustandsrückführung kompensiert werden können. Die Transformation wird erneut aus Gründen der Übersichtlichkeit am Beispiel des SISO-Systems

aus Gleichung 5.9 vorgestellt. Ausführliche Darstellung der Transformation für ein MIMO-System finden sich unter anderem in [46, 123].

Mit dem Wissen, dass bei der Existenz eines relativen Grades r für die Lie-Ableitungen $L_g \cdot L_f^k \cdot h(\vec{x}) = 0$ gilt, solange $k < r - 1$ ist, können die ersten r Koordinaten des transformierten Systems wie folgt gebildet werden:

$$\begin{aligned}
 z_1 &= \xi_1 = \Phi_1(\vec{x}) = y = L_f^0 \cdot h(\vec{x}) = h(\vec{x}) \\
 z_2 &= \xi_2 = \Phi_2(\vec{x}) = \frac{d\Phi_1}{dt} = \dot{z}_1 = \dot{y} = L_f^1 \cdot h(\vec{x}) \\
 &\vdots \\
 z_{r-1} &= \xi_{r-1} = \Phi_{r-1}(\vec{x}) = \frac{d\Phi_{r-2}}{dt} = \dot{z}_{r-2} = y^{(r-2)} = L_f^{(r-2)} \cdot h(\vec{x}) \\
 z_r &= \xi_r = \Phi_r(\vec{x}) = \frac{d\Phi_{r-1}}{dt} = \dot{z}_{r-1} = y^{(r-1)} = L_f^{(r-1)} \cdot h(\vec{x}) .
 \end{aligned} \tag{5.17}$$

Besitzt das System jedoch n Zustände, müssen weitere $n - r$ Koordinaten für das transformierte System gefunden werden, so dass sowohl die Transformationsabbildung Φ als auch dessen Inverse Φ^{-1} kontinuierlich differenzierbar sind und es sich somit bei $\vec{z} = \Phi(\vec{x})$ um einen Diffeomorphismus handelt [39]. Unter Beachtung der linearen Unabhängigkeit der verbleibenden $n - r$ Koordinatendifferenziale $d\Phi_i$ mit $r < i \leq n$ zu den bereits gewählten r Koordinaten, können diese wie folgt aufgestellt werden:

$$\begin{aligned}
 z_{r+1} &= \eta_1 = \Phi_{r+1}(\vec{x}) \\
 &\vdots \\
 z_n &= \eta_{n-r} = \Phi_n(\vec{x}) .
 \end{aligned} \tag{5.18}$$

Mit den Gleichungen 5.17 und 5.18 kann nun die vollständige Transformationsvorschrift angegeben werden:

$$\vec{z} = \Phi(\vec{x}) = \begin{bmatrix} z_1 \\ \vdots \\ z_r \\ z_{r+1} \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_r \\ \eta_1 \\ \vdots \\ \eta_{n-r} \end{bmatrix} \tag{5.19}$$

Dabei beschreiben die Zustände ξ_i die ersten r Koordinaten des transformierten Systems, die durch das mehrfache Lie-Ableiten der Ausgangsgleichung y berechnet werden und η_i die weiteren wählbaren $n - r$ Koordinaten zur Erzeugung des Diffeomorphismus. Um nun

zu einer vollständigen Beschreibung des neuen Systems zu kommen, werden mit Hilfe der Umkehrtransformation $\vec{x} = \Phi^{-1}(\vec{z})$ die folgenden Substitutionen aufgestellt:

$$a(\vec{z}) = L_g \cdot L_f^{r-1} \cdot h(\Phi^{-1}(\vec{z})) \quad \text{und} \quad b(\vec{z}) = L_f^r \cdot h(\Phi^{-1}(\vec{z})) . \quad (5.20)$$

Darüber hinaus werden i weitere Ersatzgrößen für $r+1 \leq i \leq n$ formuliert:

$$q_i(\vec{z}) = L_f \cdot \Phi_i(\Phi^{-1}(\vec{z})) \quad \text{und} \quad p_i(\vec{z}) = L_g \cdot \Phi_i(\Phi^{-1}(\vec{z})) . \quad (5.21)$$

Fasst man die einzelnen Zustände ξ_i und η_i des transformierten Systems in den beiden Teilzustandsvektoren $\vec{\xi} = [\xi_1, \dots, \xi_r]^T$ und $\vec{\eta} = [\eta_1, \dots, \eta_{n-r}]^T$ zusammen, so kann mit Hilfe der Gleichungen 5.20 und 5.21 das transformierte System in der Byrnes-Isidori-Normalform angegeben werden [38]:

$$\dot{\vec{\xi}} = \begin{bmatrix} \xi_2 \\ \vdots \\ \xi_r \\ b(\vec{\xi}, \vec{\eta}) + a(\vec{\xi}, \vec{\eta}) \cdot u \end{bmatrix}, \quad \dot{\vec{\eta}} = \vec{q}(\vec{\xi}, \vec{\eta}) + \vec{p}(\vec{\xi}, \vec{\eta}) \cdot u . \quad (5.22)$$

Dabei kann für die Ausgabe y des transformierten Systems nach Gleichung 5.17 der folgende Ausdruck angegeben werden:

$$y = h(\Phi^{-1}) = z_1 = \xi_1 . \quad (5.23)$$

Für die Transformation eines Mehrgrößensystems müssen entsprechend dem obigen Vorgehen m Teilzustandsvektoren $\vec{\xi}^i = [\xi_1^i, \dots, \xi_{r_i}^i]^T$ gefunden werden, die final zu einem kombinierten Vektor der Form $\vec{\xi}_{r \times m} = [\xi^1, \dots, \xi^m]^T$ zusammengefasst werden. Gilt für den vektoriellen Grad $r = r_1 + \dots + r_m \leq n$ müssen wiederum $n - r$ weitere Koordinaten gefunden werden, mit deren Hilfe garantiert wird, dass es sich bei $\vec{z} = \Phi(\vec{x})$ um einen Diffeomorphismus handelt. Detailliertere Beschreibung der Transformation eines Mehrgrößensystems in Normalform finden sich in [46, 123]. Mit dem Ziel einer Systemformulierung, die den Ausgleich vorhandener Nichtlinearitäten durch eine zustandsrückführende Steuergröße \vec{u} erlaubt, lässt sich nun der Vektor der jeweils r_i -ten Ableitung der Ausgangsgrößen für ein Mehrgrößensystem wie folgt angeben:

$$\left[y_i^{(r_i)} \right]_{m \times 1} = \vec{b}(\vec{x}) + \mathbf{A}(\vec{x}) \cdot \vec{u} = \vec{b}(\vec{\xi}, \vec{\eta}) + \mathbf{A}(\vec{\xi}, \vec{\eta}) \cdot \vec{u} . \quad (5.24)$$

Mit Gleichung 5.24 liegt somit ein direkter Zusammenhang zwischen der Eingangsgröße und den in der Ableitung des Ausgangs auftauchenden Nichtlinearitäten vor.

5.1.2 Linearisierende Zustandsrückführung

Unter Verwendung der im letzten Abschnitt vorgestellten Zustandstransformation kann nun eine dazugehörige Zustandsrückführung aufgestellt werden, die ein lineares Eingangs-/Ausgangsverhalten erzeugt. Dabei dient die bereits erwähnte Pseudosteuergröße

$\vec{v} = [v_1, \dots, v_m]^T$ als Eingangsgröße des transformierten Systems. Neben der linearisierenden Wirkung führt die Zustandsrückführung darüber hinaus zu einer Entkopplung der Eingangs-/Ausgangsdynamik, so dass jede Pseudosteuergröße v_i jeweils nur eine Ausgangsgröße beeinflusst. Entsprechend folgt, dass für ein System mit m Ausgangsgrößen ebenfalls m Pseudosteuergrößen definiert werden.

In Anlehnung an [46] wird die folgende nichtlineare Zustandsrückführung durch die Steuergröße \vec{u} definiert:

$$\vec{u} = \vec{\alpha}(\vec{x}) + \beta(\vec{x}) \cdot \vec{v} \quad (5.25)$$

Eingesetzt in die abgeleitete Ausgangsgleichung des transformierten Systems 5.24 ergibt sich für den geschlossenen Regelkreis:

$$\left[y_i^{(r_i)} \right]_{m \times 1} = \vec{b}(\vec{x}) + \mathbf{A}(\vec{x}) \cdot [\vec{\alpha}(\vec{x}) + \beta(\vec{x}) \cdot \vec{v}] . \quad (5.26)$$

Durch die geeignete Wahl der Koeffizienten $\vec{\alpha}(\vec{x})$ und $\beta(\vec{x})$ in Form von

$$\vec{\alpha}(\vec{x}) = -\mathbf{A}^{-1}(\vec{x}) \cdot \vec{b}(\vec{x}) \quad \text{und} \quad \beta(\vec{x}) = \mathbf{A}^{-1}(\vec{x}) \quad (5.27)$$

zerfällt die in Gleichung 5.26 gezeigte Dynamik des geregelten Systems zu:

$$\left[y_i^{(r_i)} \right]_{m \times 1} = \vec{v} . \quad (5.28)$$

Entsprechend dem oben gezeigten Beispiel ist in Gleichung 5.28 zu erkennen, dass es durch die Verwendung der linearisierenden Zustandsrückführung in Kombination mit der gezeigten Transformation möglich ist, die im System bestehenden Nichtlinearitäten im Eingangs-/Ausgangsverhalten zu kompensieren. So kann ein ursprünglich verkoppeltes nichtlineares System zu einer entkoppelten und linearen Integratorkette zwischen der Pseudosteuergröße und einer Ableitung des jeweiligen Systemausgangs vereinfacht werden. Die eigentliche Systemausgabe geht somit aus der r_i -fachen Integration der Pseudosteuergröße hervor. Mit anderen Worten kann die r_i -te Ableitung der geregelten Ausgangsgröße direkt durch die Pseudosteuergröße kommandiert werden. Der final durch die dynamische Inversion kommandierte Steuergrößenvektor \vec{u} ergibt sich durch Einsetzen der Koeffizienten aus Gleichung 5.27 in die Formulierung der Zustandsrückführung in Gleichung 5.25 zu:

$$\vec{u} = \mathbf{A}^{-1}(\vec{x}) \cdot [\vec{v} - \vec{b}(\vec{x})] . \quad (5.29)$$

Zur Veranschaulichung der Kombination von Systemtransformation und linearisierender Zustandsrückführung ist in Abbildung 5.1 ein abstraktes Blockschaltbild der dynamischen Inversion dargestellt. Zur Unterscheidung des realen zu regelnden Systems $F(\vec{x}, \vec{u})$ und des für die Inversion herangezogenen Modells $\hat{F}(\vec{x}, \vec{u})$ wird dieses mit dem Index $\hat{}$ gekennzeichnet. Entsprechend gilt für das invertierte System der Ausdruck $\hat{F}^{-1}(\vec{x}, \vec{v})$. Unter der Annahme einer idealen Inversion ergibt sich als Übertragungsverhalten von Pseudosteuergröße \vec{v} zur r_i -fach abgeleiteten Systemantwort $\vec{y}_i^{(r_i)}$ ein linearisiertes Systemverhalten von $F(s) = 1$.

Damit vereinfacht sich die ehemals nichtlineare Systemdynamik wie in Gleichung 5.28 zu sehen und in Abbildung 5.1 dargestellt, zu einer linearen Integratorkette.

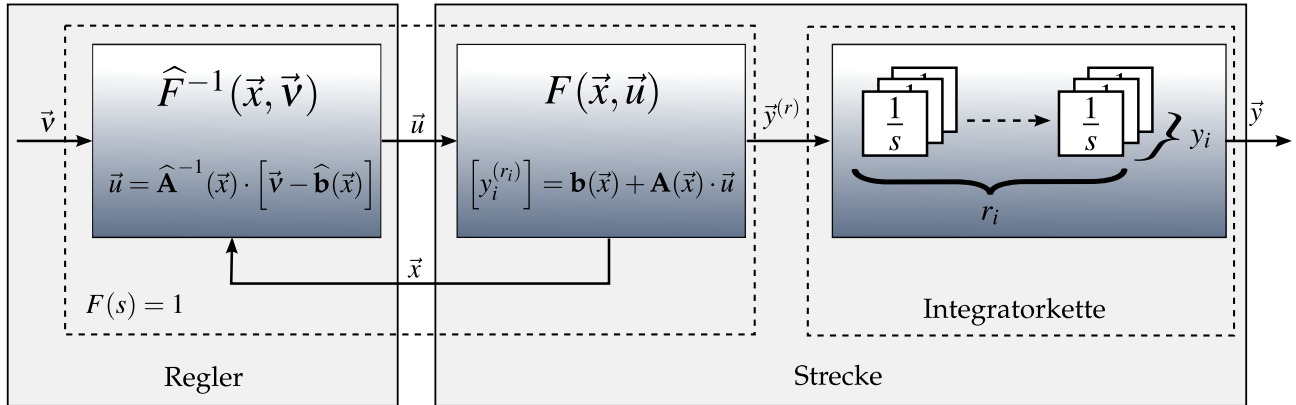


Abbildung 5.1: Abstrakte Darstellung des transformierten Systems mit linearisierender Zustandsrückführung nach [46]

Durch die Systemtransformation kommt es dabei zur Verschiebung von Zuständen in die sogenannte interne Dynamik. Betroffen davon sind die $n - r$ zusätzlichen Koordinaten des Teilzustandsvektors $\vec{\eta}$ aus Gleichung 5.22. Als Resultat dessen ist die Dynamik dieser Zustandsgrößen durch die Transformation des Systems nicht länger im Eingangs-/Ausgangsverhalten zu beobachten [38, 39]. So kann es beispielsweise bei Systemen mit instabiler interner Dynamik zu einem kontinuierlichen Anwachsen der Zustände der inneren Dynamik kommen, obwohl die Eingangs-/Ausgangsdynamik stabil und gleichbleibend geregelt wird. Aus diesem Grund liegt der Schluss nahe, dass die interne Dynamik eines mittels dynamischer Inversion geregelten Systems stabil oder zumindest beschränkt sein muss. Zur Untersuchung dieser beiden Forderungen lässt sich unter anderem das Prinzip der Minimalphasigkeit von linearen auf nichtlineare Systeme übertragen, indem es auf eine Sonderform der internen Dynamik, der Nulldynamik angewendet wird [124]. Bei dieser Sonderform werden die Zustände des Vektors $\vec{\xi}$ über die Steuergröße \vec{u} gleich Null gehalten, so dass es zu keinerlei Veränderung der externen Dynamik $\vec{\xi}(t)$ mehr kommt und die Nulldynamik entkoppelt betrachtet werden kann. So kann nach [38] und [39] die asymptotische Stabilität der Nulldynamik als hinreichende Bedingung für die asymptotische Stabilisierbarkeit des Gesamtsystems respektive für die Beschränktheit der Zustände der internen Dynamik angesehen werden. Ausführliche Betrachtungen der internen Dynamik und ihrer Stabilisierbarkeit finden sich in [38, 39, 46, 124].

5.1.3 Referenzgrößen und Inversionsfehler

Für die Erzeugung der benötigten Pseudosteuergrößen \vec{v} , die letztendlich zur Einhaltung eines vorgegebenen Referenzverlaufs $\vec{y}_R(t)$ führen, werden für gewöhnlich lineare Referenzmodelle verwendet. Mit diesen wird für jede der i gewünschten Ausgangsgrößen $y_{R,i}(t)$ die dazugehörige r_i -te Ableitung $y_{R,i}^{(r_i)}(t)$ gebildet, die anschließend als Pseudosteuergrößen in

das transformierte System gegeben werden. Über die Pseudosteuergröße hinaus wird im Referenzmodell jede weitere der $r - 1$ Ableitungen sowie die eigentliche Referenzvorgabe $y_{R,i}(t)$ berechnet. So ist direkt ersichtlich, dass die Ordnung des verwendeten Referenzmodells vom relativen Grad des geregelten Systems abhängt, da sichergestellt werden muss, dass die benötigte Anzahl von Ableitungen der Referenzausgabe vorhanden ist. In Abbildung 5.2 ist ein beispielhaftes Referenzmodell zweiter Ordnung für ein System mit relativem Grad zwei dargestellt.

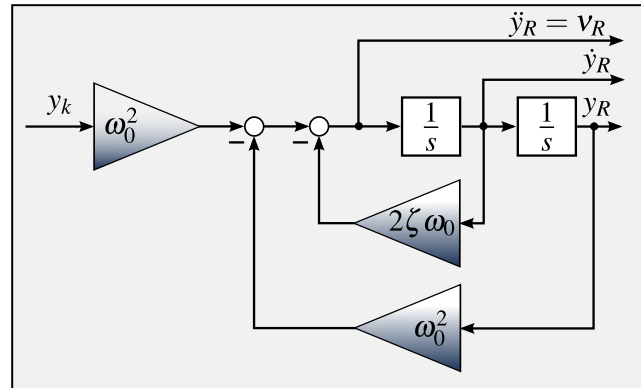


Abbildung 5.2: Lineares Referenzmodell zweiter Ordnung

Es ist zu erkennen, dass neben der Erzeugung des Referenzausgangs y_R die beiden ersten Ableitungen gebildet werden. Dabei handelt es sich bei y_k um die kommandierte Führungsgröße. Durch die Vorgabe von Dämpfungen ζ und ungedämpfter Eigenkreisfrequenz ω_0 ist es obendrein möglich, dem vorgegebenen Referenzverlauf eine Wunschdynamik aufzuprägen. Aus diesem Grund wird das Referenzmodell in der Literatur oftmals auch als Kommandofilter bezeichnet [34, 116]. So ist es möglich, sprunghafte oder von der Dynamik zu schnell kommandierte Verläufe auf ein durch das System zu erreichendes Niveau zu reduzieren. Dabei ist es leicht nachvollziehbar, dass es nicht sinnvoll ist, das Referenzmodell schneller als das eigentliche Systemverhalten zu parametrisieren.

Die im vorherigen Abschnitt beschriebene Annahme einer idealen Nachmodellierung des realen Systems, so dass die Pseudosteuergröße exakt von der Strecke umgesetzt werden kann und folglich $y_i^{(r_i)} = y_{R,i}^{(r_i)}$ gilt, ist in der Realität aus vielfältigen Gründen nicht haltbar. So führen beispielsweise unmodellerte Dynamiken, Parameterunsicherheiten, fehlerbehaftete Messgrößen, Verzögerungen und jegliche nicht messbare äußere Störungen zu Abweichungen zwischen dem simulierten Streckenverhalten $\hat{F}(\vec{x}, \vec{u})$ und dem tatsächlichen Systemverhalten $F(\vec{x}, \vec{u})$. Mit dem Ziel einer formelmäßigen Beschreibung eines solchen, durch die Verwendung der dynamischen Inversion erzeugten kombinierten Fehlers, kann das transformierte System unter Berücksichtigung der mit $\hat{\cdot}$ markierten Modellgrößen analog Gleichung 5.15 wie folgt angegeben werden:

$$\left[\hat{y}_i^{(r_i)} \right]_{m \times 1} = \hat{b}(\vec{x}) + \hat{A}(\vec{x}) \cdot \vec{u}. \quad (5.30)$$

Mit Hilfe von Gleichung 5.26 lässt sich nun der Fehler zwischen Modellannahme und Realität als sogenannter Inversionsfehler $\vec{\Delta}(\vec{x}, \vec{u})$ angeben:

$$\vec{\Delta}(\vec{x}, \vec{u}) = \begin{bmatrix} y_i^{(r_i)} \end{bmatrix}_{m \times 1} - \begin{bmatrix} \hat{y}_i^{(r_i)} \end{bmatrix}_{m \times 1} = \begin{bmatrix} \vec{b}(\vec{x}) - \hat{\vec{b}}(\vec{x}) \end{bmatrix} + \begin{bmatrix} \mathbf{A}(\vec{x}) - \hat{\mathbf{A}}(\vec{x}) \end{bmatrix} \cdot \vec{u}. \quad (5.31)$$

Somit wird der Inversionsfehler als Differenz der Systemdefinition von Realität und Modell ausgedrückt, was sich ebenfalls in dem Unterschied der r_i -ten Ableitung von y und \hat{y} niederschlägt. Eine andere Darstellung erfolgt unter der Ausnutzung der im Regler verwendeten Zustandsrückführung der Form:

$$\vec{u} = \hat{\mathbf{A}}^{-1}(\vec{x}) \cdot \begin{bmatrix} \vec{v} - \hat{\vec{b}}(\vec{x}) \end{bmatrix}. \quad (5.32)$$

So ergibt sich durch Einsetzen von Gleichung 5.32 in Gleichung 5.30 und dem Wissen, dass es sich bei der r_i -ten Ableitung der Referenzausgabe um die Pseudosteuergröße handelt, der folgende Ausdruck:

$$\begin{bmatrix} \hat{y}_i^{(r_i)} \end{bmatrix} = \hat{\vec{b}}(\vec{x}) + \hat{\mathbf{A}}(\vec{x}) \cdot \vec{u} = \hat{\vec{b}}(\vec{x}) + \hat{\mathbf{A}}(\vec{x}) \cdot \hat{\mathbf{A}}^{-1}(\vec{x}) \cdot \begin{bmatrix} y_{R,i}^{(r_i)} - \hat{\vec{b}}(\vec{x}) \end{bmatrix} = \begin{bmatrix} y_{R,i}^{(r_i)} \end{bmatrix}. \quad (5.33)$$

Damit entspricht die r_i -te Ableitung der Modellausgabe der entsprechenden r_i -ten Ableitung der Referenzvorgabe. Mit diesem Wissen ist es nun möglich, Gleichung 5.31 noch einmal umzuschreiben, so dass für den Inversionfehler $\vec{\Delta}$

$$\vec{\Delta} = \begin{bmatrix} y_i^{(r_i)} \end{bmatrix} - \begin{bmatrix} \hat{y}_i^{(r_i)} \end{bmatrix} = \begin{bmatrix} y_i^{(r_i)} \end{bmatrix} - \begin{bmatrix} y_{R,i}^{(r_i)} \end{bmatrix} \quad (5.34)$$

folgt. Damit handelt es sich beim Inversionsfehler also um die Abweichungen zwischen den Ableitungen des Referenz- und des tatsächlichen Ist-Verlaufs. Durch die r_i -fache Integration der somit fehlerbehafteten Ausgangsgröße $y_i^{(r_i)}$ kommt es dabei zu einer Fehlerfortpflanzung über sämtliche Ableitungen bis hin zur letztendlichen Ausgangsgröße y_i . Bezogen auf das eingangs-/ausgangslinearisierte Ersatzsystem lässt sich der Inversionsfehler als einwirkende Störgröße eines linearen Regelungssystems interpretieren [125]. Wie oben bereits geschildert, werden innerhalb dieser Arbeit die vorgestellten künstlichen neuronalen Netzwerke dafür verwendet, den Einfluss des Inversionsfehlers $\vec{\Delta}$ auf das System zu reduzieren, indem sie den Fehler approximieren und ausgleichen. Dabei ist es den Netzwerken über das Anlernen vorhandener Defizite der Modellbildung hinaus möglich, die Fehlertoleranz des geregelten Systems zu erhöhen, indem sie Dynamikänderungen als Folge von Systemschäden ebenfalls kompensieren.

5.1.4 Linearregler der Fehlerdynamik

In diesem Abschnitt wird auf die Abweichung zwischen den Referenzverläufen samt ihren Ableitungen und den jeweiligen tatsächlichen Systemantworten eingegangen. Dabei wird mittels eines Linearreglers die Dynamik vorgegeben, die zum Abklingen des Regelfehlers und seiner Ableitungen führt. Aus Gründen der Übersichtlichkeit wird das Vorgehen erneut

am Beispiel der exakten Eingangs-/Ausganglinearisierung eines SISO-Systems demonstriert. Entgegen der für lineare Regelungssysteme oftmals verwendeten Nomenklatur wird der Fehler für eingangs-/ausganglinearisierte Systeme in der Literatur für gewöhnlich zu

$$\vec{e} = \begin{bmatrix} e_1 \\ \vdots \\ e_m \end{bmatrix} = \begin{bmatrix} y_1 - y_{R,1} \\ \vdots \\ y_m - y_{R,m} \end{bmatrix} \quad (5.35)$$

bestimmt [38, 39, 46]. Entsprechend Gleichung 5.35 lassen sich die Ableitungen des Fehlers mit $e_i^{(j)} = y_i^{(j)} - y_{R,i}^{(j)}$ angeben. Die Aufschaltung des im Anschluss gezeigten Fehlerausgleichs erfolgt am Eingang des transformierten Systems als Teil der kommandierten Pseudosteuergröße v . Es folgt somit, dass neben der höchsten Ableitung des Referenzsignals $y_R^{(r)}$ die ersten $r - 1$ Ableitungen des Regelfehlers inklusive der Nullableitung für die Erzeugung der Pseudosteuergröße verwendet werden. Mit den reellen Regelungskoeffizienten c_i folgt:

$$v = y_R^{(r)} - c_0 \cdot (y - y_R) - c_1 \cdot (\dot{y} - \dot{y}_R) - \dots - c_{r-1} \cdot (y^{(r-1)} - y_R^{(r-1)}) . \quad (5.36)$$

Damit setzt sich die Pseudosteuergröße v also aus der im Referenzmodell gebildeten r -ten Ableitung des Referenzsignals v_R und dem Anteil der linear geregelten Fehlerrückführung v_e zusammen. Zusammengefasst lässt sich somit schreiben:

$$v = y_R^{(r)} - \sum_{i=0}^{r-1} c_i \cdot e^{(i)} = v_R + v_e . \quad (5.37)$$

Um nun die Dynamik der Regelabweichung und deren Abklingverhalten beschreiben zu können, wird der Fehlervektor $\vec{\chi}(t)$ eingeführt:

$$\vec{\chi}(t) = \left[e(t), \dot{e}(t), \dots, e^{(r-1)}(t) \right]^T . \quad (5.38)$$

Unter der Verwendung von $e^{(r)} = y^{(r)} - y_R^{(r)}$ lässt sich im Anschluss eine lineare Differentialgleichung für das Abklingverhalten der Regelabweichungen angeben, die folgerichtig als Fehlerdynamik bezeichnet wird:

$$e^{(r)} + c_{r-1} \cdot e^{(r-1)} + \dots + c_1 \cdot \dot{e} + c_0 \cdot e = e^{(r)} + \vec{c}^T \cdot \vec{\chi} = 0 . \quad (5.39)$$

Durch die Wahl des Koeffizientenvektors \vec{c} lässt sich die Dynamik dabei frei vorgeben. Einzige Voraussetzung ist, dass das Polynom $s^r + c_{r-1} \cdot s^{(r-1)} + \dots + c_1 \cdot s + c_0 = 0$ einen negativen Realteil besitzt. Die daraus resultierende Stabilität und Beschränktheit des geschlossenen Regelkreises und ihre Übertragbarkeit auf Mehrgrößensysteme sind unter anderem in [38] dargestellt. Soll nun der Inversionsfehler Δ wie oben erwähnt als Störung innerhalb der Fehlerdynamik berücksichtigt werden, kann dieser unter Umstellung der Gleichung 5.34 zu $y_i^{(r_i)} = y_{R,i}^{(r_i)} + \Delta_i$ als Eingangsgröße der Fehlerdynamik interpretiert werden:

$$e^{(r)} + \vec{c}^T \cdot \vec{\chi} = \Delta . \quad (5.40)$$

Umgeformt in die zugehörige und regelungstechnisch leicht zu interpretierende Zustandsraumdarstellung lautet die Fehlerdynamik des eingangs-/ausgangslinearisierten Systems unter Verwendung des Inversionsfehlers und Fehlerzustandsvektors:

$$\dot{\vec{\chi}} = \mathbf{A}_E \cdot \vec{\chi} + \vec{b}_E \cdot \Delta, \quad (5.41)$$

mit

$$\mathbf{A}_E = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -c_0 & -c_1 & -c_2 & \cdots & -c_{r-1} \end{bmatrix} \quad \text{und} \quad \vec{b}_E = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \quad (5.42)$$

Ein für die Stabilität der Fehlerdynamik wesentlicher Aspekt ist die Existenz einer oberen Schranke des Betrags des Fehlerzustandsvektors $|\vec{\chi}|$ unter der Annahme eines maximal auftretenden Inversionsfehlers $\bar{\Delta}$. Unter Verwendung der Ljapunow-Methode wird in [46] dargestellt, dass eine Kandidatenfunktion $\mathbf{A}_E^T \cdot \mathbf{P}_E + \mathbf{P}_E \cdot \mathbf{A}_E = -\mathbf{Q}_E$ existiert, mit deren Hilfe die obere Schranke $|\vec{\chi}|$ bestimmt werden kann.

5.1.5 Pseudo-Control Hedging

Eine Zusammenfassung der bislang vorgestellten Elemente der dynamischen Inversion ist in Abbildung 5.3 am Beispiel eines SISO-System dargestellt. Dabei werden mit der Übertragungsfunktion $G_R(s)$ das Referenzmodell und mit $G_A(s)$ der verwendete Aktuator beschrieben. Letzterer ist insofern von besonderem Interesse, als dass die Aktuatordynamik in der bisherigen Inversion explizit unberücksichtigt geblieben ist. Mit anderen Worten geht die vorgestellte Eingangs-/Ausgangslinearisierung von einem idealen und somit exakt und verzögerungsfrei reagierenden Aktuator aus, so dass das dynamische Verhalten der angesteuerten Stellglieder nicht mit invertiert wird.

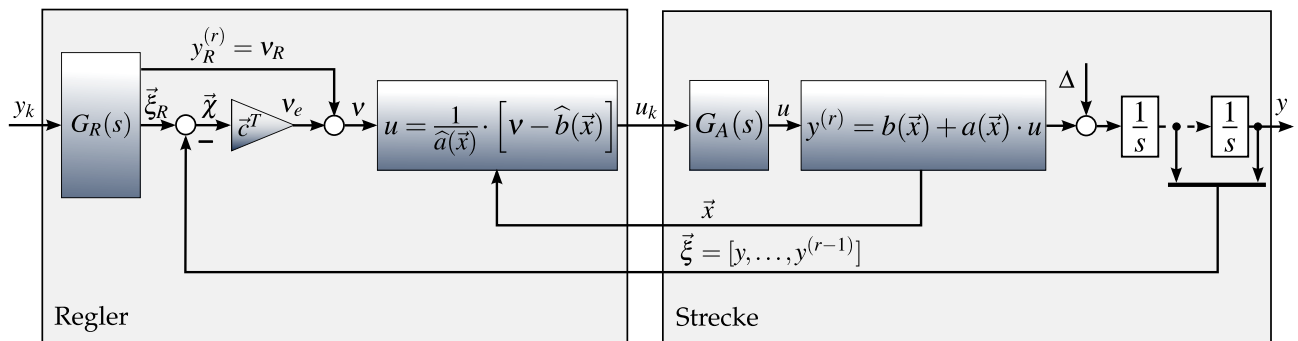


Abbildung 5.3: Struktur eines Inversionsreglers für ein Eingrößensystem nach [46]

Sowohl das dynamische Verhalten der Aktuatorik als auch mögliche Raten- und Ausschlagsbegrenzungen führen in der Realität jedoch dazu, dass die von der Inversion

kommandierte Steuergröße \vec{u}_k nicht der tatsächlichen, die Strecke beeinflussenden Steuergröße \vec{u} entspricht. Eine Möglichkeit diese limitierte Dynamik zu berücksichtigen, wäre eine ebenfalls vollständige Inversion der Aktuatordynamik. Allerdings würde sich dadurch der relative Grad des zu invertierenden Systems weiter erhöhen, was zu Gunsten einer geringeren Komplexität vermieden werden soll. Ein alternatives Vorgehen erlaubt hingegen das sogenannte Pseudo-Control Hedging (PCH) [32, 36, 37, 46, 116, 123]. Bei dieser Methode werden auftretende Verzögerungen und Begrenzungen direkt in den Referenzmodellen berücksichtigt, so dass es zu einer Reduktion der Referenzverlaufsdynamik kommt. Auf diese Weise wird sichergestellt, dass der Inversionsregler nur eine solche Wunschkynamik kommandiert, die durch die Regelstrecke auch umgesetzt werden kann.

Insbesondere bei Ausschlagsbegrenzungen, die zu einem anhaltenden Fehler zwischen der kommandierten Referenzgröße und der durch das System erreichbaren Antwort führen, kann das PCH das „Volllaufen“ von Integratoren im linearen Regler (engl. *integrator wind-up*) verhindern. Dies wird möglich, in dem das PCH den durch die Begrenzung entstehenden Fehler vor der Regelung „versteckt“. Einen ähnlichen Effekt hat das PCH auf die im Verlauf der Arbeit gezeigte Erweiterung des Inversionsreglers mit neuronalen Netzwerken. Um ein mögliches Antrainieren der Netzwerke an eine gesättigte Aktuatordynamik zu verhindern, wird der durch die Begrenzung erzeugte Fehler vor dem neuronalen Netzwerk verborgen. Andernfalls kann es zu einem konstanten Wachstum der Gewichte aufgrund der durch die Sättigung verlorengegangenen Steuerautorität kommen.

Umgesetzt wird das Pseudo-Control Hedging, indem die kommandierte Pseudosteuergröße v mit einer geschätzten Systemantwort $\hat{v} = \hat{F}(\vec{x}, u)$ verglichen und die Differenz v_h direkt im Referenzmodell abgezogen wird. Für das sogenannte Hedging-Signal eines Eingrößensystems folgt mit $u_k = \hat{F}^{-1}(\vec{x}, v)$ als Ausgang der dynamischen Inversion:

$$v_h = v - \hat{v} = \hat{F}(\vec{x}, u_k) - \hat{F}(\vec{x}, u) = v - \hat{F}\left(\vec{x}, G_A\left(\hat{F}^{-1}(\vec{x}, v)\right)\right). \quad (5.43)$$

An Gleichung 5.43 ist zu erkennen, dass bei der Berechnung von v_h einzig der Unterschied zwischen u_k und u eine Rolle spielt. In beiden Fällen wird das Systemmodell \hat{F} verwendet, so dass eine Differenz von v und \hat{v} nur durch die Aktuatordynamik G_A hervorgerufen wird. Die über das Hedging-Signal quantifizierte Verzögerung der Aktuatordynamik wird nun an das Referenzmodell zurückgeführt und dort von der r -ten Ableitung $y_R^{(r)}$ abgezogen.

$$y_R^{(r)} = v_R - v_h \quad (5.44)$$

Dargestellt ist das um das Hedging-Signal erweiterte Referenzmodell zweiter Ordnung in Abbildung 5.4. In [46] wird gezeigt, dass das Pseudo-Control Hedging keine unmittelbare Auswirkung auf die Fehlerdynamik hat, sodass weiterhin Gleichung 5.40 für die Fehlerdynamik eines Systems mit PCH gilt. Allerdings gilt es zu berücksichtigen, dass trotz der unveränderten Fehlerdynamik eine weitere Rückführung im System geschlossen wird, die in diesem Fall ein modelliertes Streckenverhalten in das Referenzmodell rückkoppelt.

5.2 Erweiterung durch neuronale Netzwerke

Wie bereits beschrieben, werden die neuronalen Netzwerke dafür genutzt, den Inversionsfehler Δ bei der Umsetzung einer Regelung auf Basis dynamischer Inversion zu approximieren und auf diese Weise dessen Auswirkung auf das Regelungssystem zu reduzieren. Über den Ausgleich gleichbleibender Modellierungsfehlern hinaus, ist es den Netzwerken im gewissen Maße möglich, auftretende Systemschäden durch schnelles Adaptieren zu kompensieren und so die Fehlertoleranz des Regelungssystems zu erhöhen. Dabei sind die in Kapitel 4 vorgestellten Trainingsverfahren durch ihre Überführung der Netzwerkzustände in den Gleitzustand besonders für ein schnelles und robustes Adaptieren des Inversionsfehlers geeignet. Nach [48] wird die Stabilitätsbetrachtung des Gesamtsystems durch das Gleitzustandstraining insofern vereinfacht, als dass die auf Grundlage der dargestellten Stabilitätsforderung berechnete Lernrate ebenfalls zur Stabilisierung der durch das Netzwerk beeinflussten Fehlerdynamik führt.

Vorteilhaft an der im Folgenden vorgestellten Regelungsarchitektur ist darüber hinaus, dass das neuronale Netzwerk nicht in Abhängigkeit der Art des existierenden Inversionsfehlers verschieden verschaltet werden muss. Dem neuronalen Netzwerk wird über die gesamte Betriebszeit das gleiche aus dem Linearregler kommende Fehlersignal als Trainingsgröße präsentiert. Auf diese Weise ist das Training unabhängig von der Ursache des Inversionsfehlers und gleicht durch nur einen Trainingsprozess alle, einen Einfluss auf die eigentliche Regelgröße habende Modellabweichungen, aus. Dabei ist es nicht von Relevanz, ob es sich letztendlich um einen Parameter-, Modell- oder Systemfehler handelt. Möglich wird dies, indem mit dem Ausgang des neuronalen Netzwerks v_{ad} , wie zuvor schon mit dem Ausgang des Linearreglers v_e , die Pseudosteuergröße v erweitert wird:

$$v = v_R + v_e - v_{ad} - v_r . \quad (5.46)$$

Für den durch das Netzwerk beigesteuerten adaptiven Pseudosteuergrößenanteil v_{ad} gilt:

$$v_{ad} = \mathbf{w}^{(2)} \cdot \vec{f}^{(2)}(\mathbf{w}^{(1)} \cdot \vec{x}) . \quad (5.47)$$

Dabei handelt es sich gemäß Kapitel 2 bei \vec{x} um den Eingangsvektor, bei $\mathbf{w}^{(1)}$ und $\mathbf{w}^{(2)}$ um die Gewichtsmatrizen und bei $\vec{f}^{(2)}$ um die sigmoide Übertragungsfunktion der verdeckten Schicht des Netzwerks. Wie in Gleichung 5.46 zu sehen ist, wird die Netzwerkausgabe v zusätzlich um den sogenannten robustifizierenden Term v_r erweitert [110]. Dieser ist für die im Verlauf folgende Stabilitäts- und Beschränktheitsanalyse von Bedeutung und kann als eine Art proportionale Fehlerrückführung verstanden werden. Bei dieser hängt die Verstärkung jedoch von der Frobeniusnorm, also der Wurzel aus der Quadratsumme aller Elemente der Gesamtgewichtsmatrix \mathbf{w} ab:

$$v_r = [k_{r0} + k_{r1} \cdot (\|\mathbf{w}\|_F + \bar{\mathbf{w}}_*)] \cdot \vec{\zeta} , \quad (5.48)$$

mit

$$\|\mathbf{w}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n w_{ij}^2} . \quad (5.49)$$

In Gleichung 5.48 handelt es sich bei $\vec{\zeta}$ um die sogenannte gefilterte Fehlergröße, die weiter unten für den Stabilitätsbeweis noch detaillierter beschrieben wird. Die beiden Entwurfsparmeter k_{r_0} und k_{r_1} sind ebenfalls für die Stabilitäts- und Beschränktheitsuntersuchung des erweiterten Regelungssystems von Bedeutung und werden in [46] ausführlich hergeleitet. Als letzten für die Gesamtstabilität wichtigen Term beschreibt $\bar{\mathbf{w}}_*$ die obere Schranke der optimalen Verbindungsgewichte und kann als Frobeniusnorm formuliert werden:

$$\|\mathbf{w}_*\|_F \leq \bar{\mathbf{w}}_* \quad (5.50)$$

Das Blockschaltbild des um ein neuronales Netzwerk erweiterten Systems ist in Abbildung 5.6 dargestellt.

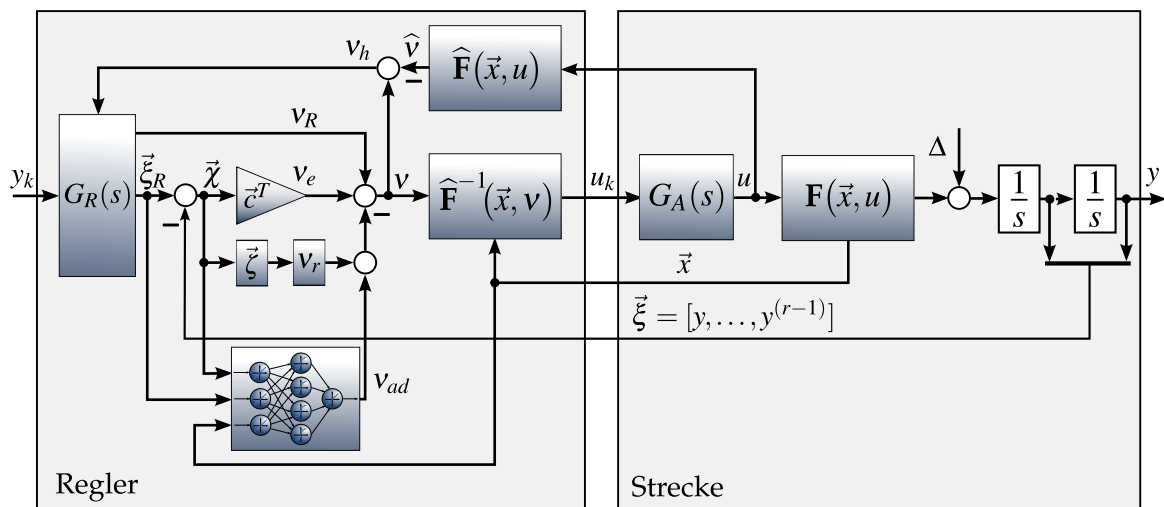


Abbildung 5.6: Struktur eines Inversionsreglers für ein Eingrößensystem mit PCH und neuronalem Netzwerk nach [46]

Für die Erweiterung des Inversionsreglers gilt es nach [46] zwei Forderungen zu erfüllen, so dass garantiert werden kann, dass der zu kompensierende Inversionsfehler durch die Netzwerkausgabe auch tatsächlich jeden Zeitschritt kleiner wird, so dass $|\partial\Delta/\partial v_{ad}| < 1$ gilt. Zum einen handelt es sich dabei um die Gleichheit von der im Modell angenommenen und der realen Wirkrichtung der Steuergrößen. Zum anderen muss die im Modell angenommene Wirkung einer Steuergröße mindestens der Hälfte der Wirkung der realen Steuergröße entsprechen und darf zu keiner Zeit als wirkungslos angenommen werden. Diese beiden Voraussetzungen können bei der Modellbildung für einen Inversionsregler sehr schnell und einfach erfüllt werden.

Eine weitere ingenieurstechnische Anforderung wird an die Zusammensetzung des Eingangsvektors \vec{x} gestellt. So müssen die Eingänge in solcher Art gewählt werden, dass sie die zu approximierende Dynamik sinnvoll beschreiben [46]. Darüber hinaus ist es üblich, den Fehlervektor $\vec{\chi}$ sowie die Referenzgrößen $\vec{\xi}_R(t)$ in das Netzwerk zu geben und nach [48] noch zeitverzögerte Eingangsgrößen zu wählen, um so die Zeitabhängigkeit der betrachteten Dynamik zu berücksichtigen.

5.2.1 Fehlerdynamik des erweiterten Systems

Durch die Erweiterung des Regelkreises mit dem neuronalen Netzwerk und dem zusätzlichen robustifizierenden Term ändert sich die Beschreibung der Fehlerdynamik aus Gleichung 5.41 entsprechend zu

$$\dot{\vec{\chi}} = \mathbf{A}_E \cdot \vec{\chi} + \vec{b}_E \cdot (\Delta - v_{ad} - v_r) . \quad (5.51)$$

Unter Verwendung von $\mathbf{w}_*^{(1)}$, $\mathbf{w}_*^{(2)}$ und \mathbf{w}_* für die Bezeichnung der optimalen Gewichte und dem Vorsatz des Inversionsfehlerausgleich $\Delta - v_{ad}$ durch das neuronale Netzwerk, lässt sich ein Restfehler der Approximation ε_A angeben, der wie folgt definiert wird:

$$\Delta - v_{ad} = \mathbf{w}_*^{(2)} \cdot \vec{f}(\mathbf{w}_*^{(1)} \cdot \vec{x}) - \mathbf{w}^{(2)} \cdot \vec{f}(\mathbf{w}^{(1)} \cdot \vec{x}) + \varepsilon_A(\vec{x}) . \quad (5.52)$$

Hierbei wird die Transferfunktion der verborgenen Schicht $\vec{f}^{(2)}$ vereinfacht als \vec{f} ausgedrückt. Nach [110] ist es nun für die Untersuchung der Stabilität der Fehlerdynamik möglich, den Restfehler näher zu quantifizieren. Zu diesem Zweck wird eine Taylorreihenentwicklung um die aktuellen Gewichte vorgenommen, die nach dem linearen ersten Glied abgebrochen wird. Auf dieser Grundlage wird anschließend die Netzwerkausgabe für die optimalen Gewichte \mathbf{w}_* bestimmt. Für die Reihenentwicklung folgt:

$$\vec{f}(\mathbf{w}_*^{(1)} \cdot \vec{x}) = \vec{f}(\mathbf{w}^{(1)} \cdot \vec{x}) + \mathbf{f}'(\mathbf{w}^{(1)} \cdot \vec{x}) \cdot [\mathbf{w}_*^{(1)} \cdot \vec{x} - \mathbf{w}^{(1)} \cdot \vec{x}] + O \left[\mathbf{w}_*^{(1)} \cdot \vec{x} - \mathbf{w}^{(1)} \cdot \vec{x} \right]^2 . \quad (5.53)$$

Für eine übersichtlichere Darstellung kann ein sogenannter Gewichtsfehler $\tilde{\mathbf{w}}$ als Unterschied zwischen den aktuellen und den optimalen Gewichten angeben:

$$\tilde{\mathbf{w}}^{(1)} = \mathbf{w}_*^{(1)} - \mathbf{w}^{(1)} , \quad \tilde{\mathbf{w}}^{(2)} = \mathbf{w}_*^{(2)} - \mathbf{w}^{(2)} , \quad \tilde{\mathbf{w}} = \mathbf{w}_* - \mathbf{w} . \quad (5.54)$$

Durch Anwendung von Gleichung 5.54 auf Gleichung 5.53, kann diese wie folgt vereinfacht werden:

$$\vec{f}(\mathbf{w}_*^{(1)} \cdot \vec{x}) = \vec{f}(\mathbf{w}^{(1)} \cdot \vec{x}) + \mathbf{f}'(\mathbf{w}^{(1)} \cdot \vec{x}) \cdot [\tilde{\mathbf{w}}^{(1)} \cdot \vec{x}] + \vec{O} \left[\tilde{\mathbf{w}}^{(1)} \cdot \vec{x} \right]^2 . \quad (5.55)$$

Dabei beschreibt $\vec{O} \left[\tilde{\mathbf{w}}^{(1)} \cdot \vec{x} \right]^2$ den durch die Linearisierung resultierenden Fehler. Gezieltes Umformen in Anlehnung an das Vorgehen in [46] führt zu einer Beschreibung des Restfehlers in Abhängigkeit des Gewichtsfehlers und der aktuellen Gewichtsmatrizen zusammen mit der Angabe eines Störterms \mathbf{z} . Es folgt:

$$\Delta - v_{ad} = \mathbf{z} + \tilde{\mathbf{w}}^{(2)} \cdot \left(\vec{f}(\mathbf{w}^{(1)} \cdot \vec{x}) - \mathbf{f}'(\mathbf{w}^{(1)} \cdot \vec{x}) \cdot \mathbf{w}^{(1)} \cdot \vec{x} \right) + \mathbf{w}^{(2)} \cdot \mathbf{f}'(\mathbf{w}^{(1)} \cdot \vec{x}) \cdot \tilde{\mathbf{w}}^{(1)} \cdot \vec{x} . \quad (5.56)$$

Der Störterm \mathbf{z} , dessen Beschränktheit $\bar{\mathbf{z}}$ in [46] dargestellt ist, setzt sich dabei wie folgt zusammen:

$$\mathbf{z} = \varepsilon_A(\vec{x}) + \tilde{\mathbf{w}}^{(2)} \cdot \mathbf{f}'(\mathbf{w}^{(1)} \cdot \vec{x}) \cdot \mathbf{w}_*^{(1)} \cdot \vec{x} + \mathbf{w}_*^{(2)} \cdot \vec{O} \left[\tilde{\mathbf{w}}^{(1)} \cdot \vec{x} \right]^2 . \quad (5.57)$$

Mit Gleichung 5.57 kann nun die erweiterte Fehlerdynamik des adaptiven Systems aus Gleichung 5.51 in ihre abschließende Form gebracht werden:

$$\dot{\vec{\chi}} = \mathbf{A}_E \cdot \vec{\chi} + \vec{b}_E \cdot \left[\mathbf{z} + \tilde{\mathbf{w}}^{(2)} \cdot \left(\vec{f}(\mathbf{w}^{(1)} \cdot \vec{x}) - \mathbf{f}'(\mathbf{w}^{(1)} \cdot \vec{x}) \cdot \mathbf{w}^{(1)} \cdot \vec{x} \right) + \mathbf{w}^{(2)} \cdot \mathbf{f}'(\mathbf{w}^{(1)} \cdot \vec{x}) \cdot \tilde{\mathbf{w}}^{(1)} \cdot \vec{x} - v_r \right] . \quad (5.58)$$

5.2.2 Beschränktheitsbetrachtung des adaptiven Reglers

Um die Stabilität des vorgestellten erweiterten Regelungskonzepts nachzuweisen, gilt es, die Beschränktheit der Netzwerkgewichte sowie des Regelfehlers zu garantieren. Ausführliche Darstellungen der Nachweisführung sowie die finale Beschreibung der notwendigen Grenzen sind [46, 110] zu entnehmen. Dabei ist es das generelle Vorgehen, zunächst die Ableitung einer geeigneten Ljapunow-Funktion aufzustellen, um anhand dieser die Beschränktheit der einzelnen variablen Anteile zu demonstrieren, so dass die Bedingung $\dot{V} < 0$ garantiert ist. An dieser Stelle soll darauf verzichtet werden, eine weitere umfassende Abhandlung der Berechnung aller Grenzen aufzuführen. Vielmehr soll, anhand der Aufstellung der Ljapunow-Funktion und der in Anhang B gezeigten Herleitung einer oberen Gewichtsschranke, der Transfer der in [46] gezeigten Beweisführung auf das in dieser Arbeit beschriebene Gleitzustandstraining demonstriert werden.

Für die Anwendung des in [110] und [46] gezeigten Ljapunow-Verfahrens zur Stabilitätsuntersuchung des adaptiven Reglers werden zunächst die beiden einzeln aufgestellten Gewichtsänderungen aus den Gleichungen 2.22 und 2.23 um die sogenannte e-Modifikation erweitert. Diese garantiert eine von \vec{x} und $\vec{\zeta}$ unabhängige obere Grenze der Gewichtsmatrix, so dass die Bedingung $\dot{V} < 0$ erfüllt werden kann. Im Vergleich zu [46] und [48] wird die im jeweils letzten Summanden stehende e-Modifikation noch um das Vorzeichen des gefilterten Fehlers erweitert. In Anhang B ist dargestellt, wie auf diese Weise eine obere Schranke der Gewichtsmatrix für das Gleitzustandstraining gefunden werden kann. Für das erweiterte Lerngesetz des Gleitzustandsverfahrens mit e-Modifikation folgt:

$$\dot{\mathbf{w}}^{(1)} = \mu^{(1)} \cdot \text{sign}(S_t) \cdot \left[\vec{x} \cdot |\vec{\zeta}| \cdot \mathbf{w}^{(2)T} \cdot \mathbf{f}'(\mathbf{w}^{(1)} \vec{x}) - \lambda_* \cdot \text{sign}(\vec{\zeta}) \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(1)} \right], \quad (5.59)$$

$$\begin{aligned} \dot{\mathbf{w}}^{(2)} = & \mu^{(2)} \cdot \text{sign}(S_t) \cdot \left[\vec{f}(\mathbf{w}^{(1)} \vec{x}) \cdot |\vec{\zeta}| - \mathbf{f}'(\mathbf{w}^{(1)} \vec{x}) \cdot \mathbf{w}^{(1)T} \vec{x} \cdot |\vec{\zeta}| \right. \\ & \left. - \lambda_* \cdot \text{sign}(\vec{\zeta}) \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(2)} \right]. \end{aligned} \quad (5.60)$$

Anschaulich wird die e-Modifikation auch *Term des Vergessens* genannt, da durch ihn die Netzwerkgewichte über die Zeit wieder zurück in Richtung Null gebracht werden. Der Auslegungsparameter λ_* kann dabei frei vorgegeben werden, in [46] wird jedoch vorgeschlagen, ihn möglichst niedrig zu wählen, da es nicht das Ziel sein kann, den Wissensaufbau im Netzwerk zu sehr zu hemmen. Zusätzlich findet nach [110] eine weitere Anpassung des ursprünglichen Backpropagation-Algorithmus in Form der Ergänzung um die Aktivierungsfunktionsableitung der verdeckten Schicht $\mathbf{f}'(\mathbf{w}^{(1)} \vec{x})$ in Gleichung 5.60 statt. Auf diese Weise wird dem Einfluss der in Gleichung 5.53 gezeigten Taylorreihenentwicklung bei der Beschreibung der Fehlerdynamik Rechnung getragen. Aus Gründen der Übersicht, wird die Ableitung dabei in den folgenden Umformungen verkürzt als \mathbf{f}' geschrieben.

Bei dem Ausdruck $\|\vec{\zeta}\|_2$ in den Gleichungen 5.59 und 5.60 handelt es sich um die euklidische Norm der bereits eingeführten gefilterten Fehlergröße. Dabei wird diese nach dem Zusammenhang $\vec{\zeta} = \vec{\chi}^T \cdot \mathbf{P}_E \cdot \vec{b}_E$ gebildet und steht somit stellvertretend für den bislang verwendeten Netzwerkfehler $\vec{\varepsilon}$ als Teil des Fehlervektors $\vec{\chi}$. Analog der Darstellung in Abschnitt 5.1.4 handelt es sich bei \vec{b}_E um den Eingangsvektor der linearen Fehlerdynamik, so dass die Matrix \mathbf{P}_E in der Art gewählt werden muss, dass sie die Ljapunow-Funktion $\mathbf{A}_E^T \cdot \mathbf{P}_E + \mathbf{P}_E \cdot \mathbf{A}_E = -\mathbf{Q}_E$ erfüllt. Für die in dieser Arbeit betrachteten Zusammenhänge vom relativen Grad von eins, vereinfacht sich der Ausdruck noch weiter zu $\vec{\zeta} = \varepsilon \cdot \mathbf{P}_E$, sodass der gefilterte Fehler in direkter Proportionalität zum bisher angenommenen Fehler ε steht.

Für den Beweis der schließlichen Beschränktheit wird die direkte Methode von Ljapunow [81] genutzt. Dabei wird für die Anwendung auf das Gleitzustandsverfahren mit möglichen negativen Lernraten, anders als in [110], ausschließlich der Betrag $|\mu|^{-1}$ in der Kandidatenfunktion berücksichtigt. Auf diese Weise, stets positiv formuliert, folgt:

$$V(\vec{\chi}) = \frac{1}{2} \vec{\chi}^T \mathbf{P}_E \vec{\chi} + \frac{1}{2} \text{Spur} \left\{ \tilde{\mathbf{w}}^{(1)T} |\mu^{(1)}|^{-1} \tilde{\mathbf{w}}^{(1)} \right\} + \frac{1}{2} \text{Spur} \left\{ \tilde{\mathbf{w}}^{(2)T} |\mu^{(2)}|^{-1} \tilde{\mathbf{w}}^{(2)} \right\}. \quad (5.61)$$

Um nun Stabilität nach Ljapunow zu beweisen, muss die erste Zeitableitung der Kandidatenfunktion negativ definit bzw. negativ semi-definit sein. Für \dot{V} folgt:

$$\begin{aligned} \dot{V} = & \frac{1}{2} \dot{\vec{\chi}}^T \mathbf{P}_E \vec{\chi} + \frac{1}{2} \vec{\chi}^T \mathbf{P}_E \dot{\vec{\chi}} + \frac{1}{2} \text{Spur} \left\{ \dot{\tilde{\mathbf{w}}}^{(1)T} |\mu^{(1)}|^{-1} \tilde{\mathbf{w}}^{(1)} \right\} + \frac{1}{2} \text{Spur} \left\{ \tilde{\mathbf{w}}^{(1)T} |\mu^{(1)}|^{-1} \dot{\tilde{\mathbf{w}}}^{(1)} \right\} \\ & + \frac{1}{2} \text{Spur} \left\{ \dot{\tilde{\mathbf{w}}}^{(2)T} |\mu^{(2)}|^{-1} \tilde{\mathbf{w}}^{(2)} \right\} + \frac{1}{2} \text{Spur} \left\{ \tilde{\mathbf{w}}^{(2)T} |\mu^{(2)}|^{-1} \dot{\tilde{\mathbf{w}}}^{(2)} \right\}. \end{aligned} \quad (5.62)$$

Mit dem Wissen, dass für die Änderung der Gewichtsfehler $\dot{\tilde{\mathbf{w}}}^{(1)} = -\dot{\mathbf{w}}^{(1)}$ und entsprechend $\dot{\tilde{\mathbf{w}}}^{(2)} = -\dot{\mathbf{w}}^{(2)}$ gilt, da sich die optimalen Gewichte nicht ändern, kann Gleichung 5.62 nach [46] verkürzt werden:

$$\dot{V} = \frac{1}{2} \dot{\vec{\chi}}^T \mathbf{P}_E \vec{\chi} + \frac{1}{2} \vec{\chi}^T \mathbf{P}_E \dot{\vec{\chi}} - \text{Spur} \left\{ \tilde{\mathbf{w}}^{(1)T} |\mu^{(1)}|^{-1} \dot{\mathbf{w}}^{(1)} \right\} - \text{Spur} \left\{ \tilde{\mathbf{w}}^{(2)T} |\mu^{(2)}|^{-1} \dot{\mathbf{w}}^{(2)} \right\}. \quad (5.63)$$

In diesen Ausdruck können nun die Fehlerdynamik aus Gleichung 5.51 und die Gewichtsänderungen aus den Gleichungen Gleichung 5.59 und 5.60 eingesetzt werden.

$$\begin{aligned} \dot{V} = & \frac{1}{2} \left[\mathbf{A}_E \cdot \vec{\chi} + \vec{b}_E \cdot (\Delta - \mathbf{v}_{ad} - \mathbf{v}_r) \right]^T \mathbf{P}_E \vec{\chi} + \frac{1}{2} \vec{\chi}^T \mathbf{P}_E \left[\mathbf{A}_E \cdot \vec{\chi} + \vec{b}_E \cdot (\Delta - \mathbf{v}_{ad} - \mathbf{v}_r) \right] \\ & - \text{Spur} \left\{ \tilde{\mathbf{w}}^{(1)T} |\mu^{(1)}|^{-1} \cdot \mu^{(1)} \cdot \text{sign}(S_t) \cdot \left[\vec{x} \cdot |\vec{\zeta}| \cdot \mathbf{w}^{(2)T} \cdot \mathbf{f}' - \lambda_* \cdot \text{sign}(\vec{\zeta}) \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(1)} \right] \right\} \\ & - \text{Spur} \left\{ \tilde{\mathbf{w}}^{(2)T} |\mu^{(2)}|^{-1} \cdot \mu^{(2)} \cdot \text{sign}(S_t) \cdot \left[\vec{f}(\mathbf{w}^{(1)} \vec{x}) \cdot |\vec{\zeta}| - \mathbf{f}' \cdot \mathbf{w}^{(1)T} \vec{x} \cdot |\vec{\zeta}| \right. \right. \\ & \left. \left. - \lambda_* \cdot \text{sign}(\vec{\zeta}) \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(2)} \right] \right\} \end{aligned} \quad (5.64)$$

Nach dem Rauskürzen des Lernratenbetrags, dem Einsetzen der Ausdrücke für $\Delta - v_{ad}$, v_r und $\vec{\zeta}$ sowie der Verwendung der Ljapunow-Funktion $\mathbf{A}_E^T \cdot \mathbf{P}_E + \mathbf{P}_E \cdot \mathbf{A}_E = -\mathbf{Q}_E$ kann Gleichung 5.64 weiter umgeschrieben werden.

$$\begin{aligned} \dot{V} = & -\frac{1}{2} \tilde{\mathbf{x}}^T \mathbf{Q}_E \tilde{\mathbf{x}} + \vec{\zeta} \cdot \left[\mathbf{z} + \tilde{\mathbf{w}}^{(2)T} \cdot \left[\vec{f}(\mathbf{w}^{(1)} \tilde{\mathbf{x}}) - \mathbf{f}' \cdot \mathbf{w}^{(1)T} \tilde{\mathbf{x}} \right] + \mathbf{w}^{(2)T} \cdot \mathbf{f}' \cdot \tilde{\mathbf{w}}^{(1)T} \tilde{\mathbf{x}} \right] \\ & - \vec{\zeta} \cdot \left[k_{r_0} + k_{r_1} \cdot (\|\mathbf{w}\|_F + \bar{\mathbf{w}}_*) \right] \cdot \vec{\zeta}^T \\ & - \text{Spur} \left\{ \tilde{\mathbf{w}}^{(1)T} \cdot \text{sign}(\mu^{(1)}) \cdot \text{sign}(S_t) \cdot \left[\tilde{\mathbf{x}} \cdot |\vec{\zeta}| \cdot \mathbf{w}^{(2)T} \cdot \mathbf{f}' - \lambda_* \cdot \text{sign}(\vec{\zeta}) \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(1)} \right] \right\} \\ & - \text{Spur} \left\{ \tilde{\mathbf{w}}^{(2)T} \cdot \text{sign}(\mu^{(2)}) \cdot \text{sign}(S_t) \cdot \left[\vec{f}(\mathbf{w}^{(1)} \tilde{\mathbf{x}}) \cdot |\vec{\zeta}| - \mathbf{f}' \cdot \mathbf{w}^{(1)T} \tilde{\mathbf{x}} \cdot |\vec{\zeta}| \right. \right. \\ & \left. \left. - \lambda_* \cdot \text{sign}(\vec{\zeta}) \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(2)} \right] \right\}. \end{aligned} \quad (5.65)$$

Dieser Ausdruck kann nun geschickt umgruppiert werden, so dass die identischen Klammerausdrücke der jeweils ersten Zeile und der *Spur*-Ausdrücke direkt hintereinander stehen. Es folgt:

$$\begin{aligned} \dot{V} = & -\frac{1}{2} \tilde{\mathbf{x}}^T \mathbf{Q}_E \tilde{\mathbf{x}} - \vec{\zeta} \cdot \left[k_{r_0} + k_{r_1} \cdot (\|\mathbf{w}\|_F + \bar{\mathbf{w}}_*) \right] \cdot \vec{\zeta}^T \\ & + \underbrace{\vec{\zeta} \cdot \tilde{\mathbf{w}}^{(2)T} \cdot \left[\vec{f}(\mathbf{w}^{(1)} \tilde{\mathbf{x}}) - \mathbf{f}' \cdot \mathbf{w}^{(1)T} \tilde{\mathbf{x}} \right]}_I \\ & - \text{Spur} \left\{ \underbrace{\tilde{\mathbf{w}}^{(2)T} \cdot \text{sign}(\mu^{(2)}) \cdot \text{sign}(S_t) \cdot \left[\vec{f}(\mathbf{w}^{(1)} \tilde{\mathbf{x}}) - \mathbf{f}' \cdot \mathbf{w}^{(1)T} \tilde{\mathbf{x}} \right] \cdot |\vec{\zeta}|}_I \right\} \\ & + \underbrace{\vec{\zeta} \cdot \mathbf{w}^{(2)T} \cdot \mathbf{f}' \cdot \tilde{\mathbf{w}}^{(1)T} \tilde{\mathbf{x}}}_{II} - \text{Spur} \left\{ \underbrace{\tilde{\mathbf{w}}^{(1)T} \cdot \text{sign}(\mu^{(1)}) \cdot \text{sign}(S_t) \cdot \tilde{\mathbf{x}} \cdot |\vec{\zeta}| \cdot \mathbf{w}^{(2)T} \cdot \mathbf{f}'}_{II} \right\} \\ & + \vec{\zeta} \cdot \mathbf{z} + \lambda_* \cdot \text{sign}(\mu^{(2)}) \cdot \text{sign}(S_t) \cdot \text{sign}(\vec{\zeta}) \cdot \|\vec{\zeta}\|_2 \cdot \text{Spur} \left\{ \tilde{\mathbf{w}}^{(2)T} \mathbf{w}^{(2)} \right\} \\ & + \lambda_* \cdot \text{sign}(\mu^{(1)}) \cdot \text{sign}(S_t) \cdot \text{sign}(\vec{\zeta}) \cdot \|\vec{\zeta}\|_2 \cdot \text{Spur} \left\{ \tilde{\mathbf{w}}^{(1)T} \mathbf{w}^{(1)} \right\}. \end{aligned} \quad (5.66)$$

An dieser Stelle gilt es nun die auftretenden Unterschiede des Gleitzustandstrainings zu den Ausführungen in [46] und [110] zu diskutieren. Ziel der Umgruppierung ist es, die jeweils mit *I* und *II* markierten Ausdrücke durch den Zusammenhang $\mathbf{a}^T \mathbf{b} = \text{Spur} \{ \mathbf{b} \cdot \mathbf{a}^T \}$ heraus zu kürzen, so dass ein deutlich übersichtlicher Ausdruck für \dot{V} entsteht. Dabei tritt für den hier gezeigten Beweis mit negativen Lernraten nun die Schwierigkeit auf, dass im jeweiligen *Spur*-Ausdruck von *I* und *II* das Produkt $\text{sign}(\mu) \cdot \text{sign}(S_t) \cdot |\vec{\zeta}|$ enthalten ist. Im jeweils ersten Ausdruck von *I* und *II* ist stattdessen der einfache gefilterte Fehler $\vec{\zeta}$ aufgeführt. Setzt man jedoch die in Abschnitt 4.3.3 aufgestellte Bedingung für einen abnehmenden Fehler in jedem Zeitschritt mit $\text{sign}(\mu) = \text{sign}(S_t) \cdot \text{sign}(\vec{\zeta})$ ein, folgt für das Produkt der Vorzeichenterme:

$$\text{sign}(\mu) \cdot \text{sign}(S_t) \cdot |\vec{\zeta}| = \text{sign}(S_t) \cdot \text{sign}(\vec{\zeta}) \cdot \text{sign}(S_t) \cdot |\vec{\zeta}| = \text{sign}(\vec{\zeta}) \cdot |\vec{\zeta}| = \vec{\zeta}. \quad (5.67)$$

Es kann somit gezeigt werden, dass durch die bereits getroffene Annahme der Vorzeichenkonvention für die Lernrate, die Zeitableitung der Kandidatenfunktion so vereinfacht werden kann, dass sie der Form aus [46] und [110] entspricht. Auf diese Weise sind die in der

Theorie existierenden Beweise der Beschränktheit und letztendlich der Stabilität eines mit neuronalen Netzwerken erweiterten Inversionsregler auf den hier gezeigten Ansatz mit Umsetzung eines Gleitzustandstrainings übertragbar. Unter Verwendung der Gesamtgewichtsmatrix \mathbf{w} folgt final für die Zeitableitung der Ljapunow-Funktion:

$$\begin{aligned} \dot{V} = & -\frac{1}{2}\vec{\chi}^T \mathbf{Q}_E \vec{\chi} - \vec{\zeta} \cdot [k_{r_0} + k_{r_1} \cdot (\|\mathbf{w}\|_F + \bar{\mathbf{w}}_*)] \vec{\zeta}^T + \vec{\zeta} \cdot \mathbf{z} \\ & + \lambda_* \cdot \|\vec{\zeta}\|_2 \cdot \text{Spur} \left\{ \tilde{\mathbf{w}}^T \cdot \tilde{\mathbf{w}} \right\} . \end{aligned} \quad (5.68)$$

Mit Gleichung 5.68 liegt nun also ein Ausdruck vor, mit dessen Hilfe sich Grenzen berechnen lassen, so dass die Bedingung $\dot{V} < 0$ erfüllt werden kann [33, 37, 46]. In Anhang B ist exemplarisch die Beschränktheit der Gewichtsmatrix für die Anwendung des Gleitzustandstrainings dargestellt.

Aus der Übertragbarkeit des auf der strukturvariablen Regelung basierenden Trainingsverfahrens auf die Anwendung innerhalb des erweiterten Inversionsreglers resultiert nach [48] ein erheblicher Vorteil für die Stabilität und Robustheit der Fehlerdynamik. Durch die Verwendung des Gleitzustandstrainings werden im Falle des erweiterten Reglers die Zustände der Fehlerdynamik ε und $\dot{\varepsilon}$ in Richtung des Gleitzustands geführt. Auf diese Weise kann die stabilisierende und robustifizierende Wirkung eines Gleitzustandsverfahren direkt auf die Fehlerdynamik des Inversionsreglers übertragen werden. Die Aufstellung des stabilen Intervalls zur Berechnung der Lernrate führt somit nicht nur zu einer Erhöhung der Konvergenzgeschwindigkeit, sondern darüber hinaus zur Stabilisierung des geschlossenen Regelungssystems [48].

Kapitel 6

Simulationsumgebung und Reglerumsetzung

Das im vorangegangenen Kapitel beschriebene Regelungskonzept, als beispielhafte Anwendung gleitzustandstrainierter neuronaler Netzwerke, ist ein generischer Ansatz, der auf verschiedenartige Problemstellungen angewendet werden kann. Die generelle Übertragbarkeit auf unterschiedliche Regelungsaufgaben soll in dieser Arbeit anhand zweier Umsetzungsvarianten demonstriert werden.

Bei dem ersten gezeigten Beispiel handelt es sich um die Umsetzung eines vollständigen Flugreglers für die Längs- und Seitendynamik des kleinen unbemannten Flugsystems *Carolo P360* mit einer Abflugmasse von 25kg in einer nichtlinearen Simulationsumgebung. Dabei soll durch den Einsatz dreier neuronaler Netzwerke für die Erweiterung der Drehratenregelung von p (Rollrate), q (Nickrate) und r (Gierrate) vor allem der schädliche Einfluss von Parameterunsicherheiten und die charakteristische Empfindlichkeit gegenüber atmosphärischen Störungen reduziert werden. Umfassende Beschreibungen der Regelungsarchitektur eines erweiterten Inversionsreglers für den betrachteten *P360* mit Anwendung von gleitzustandstrainierten neuronalen Netzwerken können in [48, 90, 92, 102] gefunden werden, so dass in dieser Arbeit die finale Umsetzung nur in verkürzter Form dargestellt werden soll.

Als zweite Variante eines adaptiven Inversionsreglers wird in dieser Arbeit die Querdynamikregelung eines PKWs der Kompaktklasse mit einem Gewicht von ungefähr 1500kg im Kontext des automatischen Fahrens vorgestellt. Anders als im Beispiel des Flugreglers wird ausschließlich die Seitendynamik, genauer die Gierbewegung, mittels nichtlinearer dynamischer Inversion geregelt, so dass nur ein neuronales Netzwerk eingesetzt wird. Die Längsregelung zur Einhaltung der kommandierten Geschwindigkeitsvorgabe erfolgt dabei durch einen einfachen PID-Regler, dessen Regelgüte nicht Teil der angestellten Untersuchungen ist. Ebenfalls unterschiedlich zu den ausschließlich simulativen Untersuchungen des Flugreglers ist die innerhalb dieser Arbeit beschriebene Übertragung der Fahrzeugregelung in den praktischen Fahrversuch. So kann neben der Simulation von Systemschäden, wie dem

Platzen eines Reifens und dessen Kompensation durch das neuronale Netz, ebenfalls der Ausgleich von Modellungenauigkeiten bei der Umsetzung des Reglers gegenüber dem tatsächlichen Fahrzeug untersucht werden. Insbesondere die in Abschnitt 6.2.2 beschriebene Vereinfachung des invertierten Reifenmodells motiviert den Einsatz eines schnellen und stabilen Trainingsverfahrens und bieten somit die Grundlage für die Untersuchung des Gleitzustandstrainings für das automatische Fahren.

6.1 Umsetzung als Flugregler

Wie bereits erwähnt, finden sich umfassende Beschreibungen eines umgesetzten Flugreglers auf Basis dynamischer Inversion unter anderem in [10, 46, 48, 126]. Dabei entspricht die im Anschluss gezeigte Umsetzung des Flugreglers für den *Carolo P360* und die verwendete Simulation exakt der Struktur, wie sie in der dieser Arbeit vorangegangenen Dissertation [48] beschrieben wurde.

6.1.1 Simulationsumgebung

Da die in Kapitel 7 dargestellten Ergebnisse und Bewertungen des adaptiven Flugreglers auf Grundlage von Simulationen beruhen, soll die verwendete Umgebung im Anschluss kurz dargestellt werden. Eine umfassende Beschreibung aller für die Simulation von Flugzeugen benötigter Komponenten findet sich detailliert dargestellt in [127].

Der prinzipielle Aufbau der für die Flugregelung verwendeten Simulationsumgebung ist als abstrahiertes Blockschaltbild in Abbildung 6.1 zu sehen.

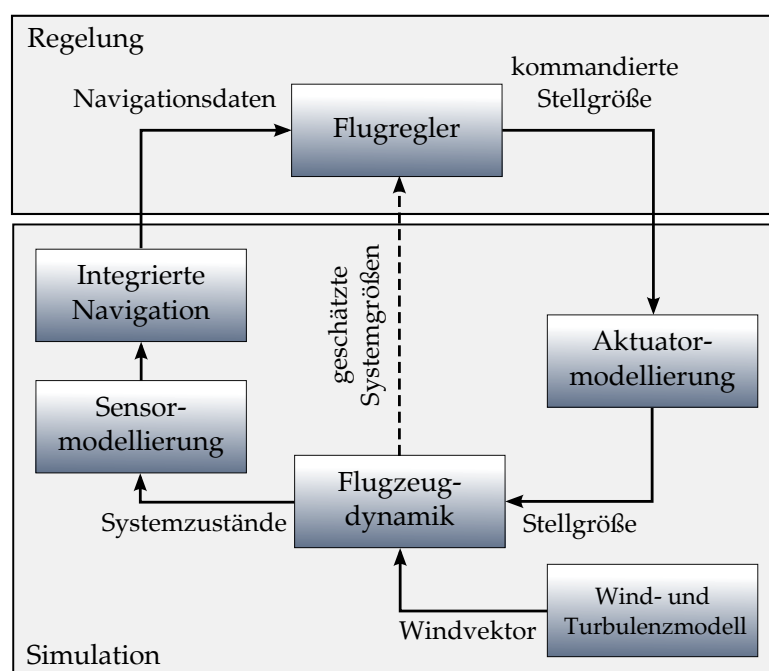


Abbildung 6.1: Blockschaltbild der nichtlinearen Simulationsumgebung

Dabei kann die vollständige Simulationsumgebung in die beiden übergeordneten Blöcke Regelung und Simulation unterteilt werden. Sind die Schnittstellen zwischen diesen beiden Strukturen eindeutig definiert, können Änderungen an der Regelung respektive an der Simulation vorgenommen werden, ohne dass die Simulationsumgebung als Ganzes überarbeitet werden muss. Zusätzlich erlaubt eine solche Struktur den einfachen Transfer in praktische Feldversuche, da es aus Sicht der Regelung keine Rolle spielt, ob die eingehenden Sensorsignale das Ergebnis einer Simulation oder realer Messungen sind. Die Beschreibung der Flugregelungsstruktur für das Beispiel eines kleinen unbemannten Flugsystems findet sich im anschließenden Abschnitt 6.1.2.

Die Güte der Simulation wird maßgeblich durch den Block Flugzeugdynamik beeinflusst. In diesem werden über die am Flugzeug herrschenden Kräfte- und Momentengleichgewichte die Zustände der Längs- und Seitenbewegung berechnet. Durch die Abbildung der aerodynamischen Eigenschaften des Flugzeugs können dessen charakteristischen Eigenbewegungsformen als ein System von Differentialgleichungen umgesetzt werden. Bevor jedoch die für die Simulation eines Flugzeugs benötigten 12 Differentialgleichungen für sechs Freiheitsgraden aufgestellt werden können, ist es nach [127] üblich einige Vereinfachungen zu treffen:

1. Es wird von einer ruhenden und ebenen Erde ausgegangen.
2. Das Flugzeug wird als Starrkörper ohne elastische Freiheitsgrade betrachtet.
3. Bezüglich der angreifenden Kräfte und Momente wird das Flugzeug als Einpunktmodell mit konstanter Masse angesehen.
4. Das Flugzeug ist symmetrisch in der x - z -Ebene.
5. Die Aerodynamik des Flugzeugs wird als quasistationär angenommen.
6. Die Kreiselkopplung des rotierenden Propellers wird vernachlässigt.

Mit diesen getroffenen Annahmen und dem flugzeugfesten Koordinatensystem als Bezugskoordinatensystem können nun die für die Beschreibung der Flugzeugdynamik verwendeten Bewegungsgleichungen aufgestellt werden. Mit der aerodynamischen Kraft \vec{R}_f^A , dem Triebwerksschube $\vec{F}_{TW,f}$ und der Gewichtskraft \vec{G}_f kann der Impulssatz für die Beschreibung der körperfesten translatorischen Geschwindigkeitskomponenten wie folgt formuliert werden:

$$\dot{\vec{V}}_{Kf}^f = \frac{1}{m} \cdot \left(\sum \vec{F} \right)_f - \vec{\omega}_f^{gf} \times \vec{V}_{Kf} = \frac{1}{m} \cdot \left(\vec{R}_f^A + \vec{F}_{TW,f} + \vec{G}_f \right)_f - \begin{bmatrix} p_k \\ q_k \\ r_k \end{bmatrix}_f \times \begin{bmatrix} u_k \\ v_k \\ w_k \end{bmatrix}_f . \quad (6.1)$$

Dabei beinhaltet die Größe $\vec{\omega}_f^{gf}$ die flugzeugfesten Drehraten gegenüber dem erdfesten System. Die aerodynamischen Kräfte werden über ein Derivatmodell mit statischer Stabilität

in Längs- und Seitendynamik berechnet. Ausführliche Berechnungen der Derivative finden sich in [118].

Für die Dynamikbeschreibung der rotatorischen Geschwindigkeitskomponenten wird der Drallsatz genutzt. Über das Momentengleichgewicht am Schwerpunkt und mit dem Trägheitstensor \mathbf{T} für ein symmetrisches Flugzeug kann die Änderung der Drehraten wie folgt geschrieben werden:

$$\dot{\vec{\omega}}_f^{gf} = \mathbf{T}^{-1} \cdot \left(\sum \vec{M}_f - \vec{\omega}_f^{gf} \times [\mathbf{T} \cdot \vec{\omega}_f^{gf}] \right) . \quad (6.2)$$

Mit den Gleichungen 6.1 und 6.2 ist es nun also möglich, die translatorische und rotatorische Bewegung des Flugzeugs im Raum zu beschreiben. Neben diesen Gleichgewichtssätzen werden für die Simulation zusätzlich noch die Lage und die Position des Flugzeugs im Raum benötigt. Für die Lage werden die Eulerwinkel als Transformation der Drehraten verwendet.

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \mathbf{M}_{\Phi f} \cdot \vec{\omega}_f^{gf} = \begin{bmatrix} 1 & \sin \Phi \cdot \tan \Theta & \cos \Phi \cdot \tan \Theta \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi / \cos \Theta & \cos \Phi / \cos \Theta \end{bmatrix}_f \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix}_f \quad (6.3)$$

Dabei gibt Φ den Hänge-, Θ den Nick- und Ψ den Gierwinkel an. Wie in der letzten Zeile von Gleichung 6.3 zu erkennen ist, kommt es bei einem Nickwinkel von $\pm 90^\circ$ zu einer Singularität, die durch die Nutzung von Quaternionen für die Berechnung der Eulerwinkel behoben wird. Für die Beschreibung des aktuellen Positionsvektors \vec{s}_g als Änderung gegenüber dem initialen Startpunkt, wird der Bahngeschwindigkeitsvektor \vec{V}_{KF} aus Gleichung 6.1 integriert und mit der Transformationsmatrix \mathbf{M}_{gf} vom flugzeugfesten in das geodätische Koordinatensystem transformiert. Es folgt somit:

$$\frac{d(\vec{s}_g)}{dt} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_g = \begin{bmatrix} u_K \\ v_k \\ w_k \end{bmatrix} = \mathbf{M}_{gf} \cdot \vec{V}_{KF} = \mathbf{M}_{gf} \cdot \begin{bmatrix} u_K \\ v_k \\ w_k \end{bmatrix} . \quad (6.4)$$

Mit den Gleichungen 6.1 bis 6.4 sind nun alle 12 benötigten Differentialgleichungen des Blocks Flugdynamik für die Simulation des *Carolo P360* dargestellt. Die Simulationsgüte, als Vergleich des mathematischen Modells gegenüber dem realen Flugverhalten, wird dabei maßgeblich durch die Qualität der identifizierten aerodynamischen Parameter beeinflusst. Da das verwendete Simulationsmodell auch immer Ausgangspunkt der umgesetzten Inversionsregelung ist, führt ein mit großen Parameterunsicherheiten umgesetztes Simulationsmodell auch immer zu einer fehlerhaften modellbasierten Regelung, wenn sie auf das tatsächliche Flugzeug angewendet wird. Wie eingangs dargestellt, motiviert diese Tatsache maßgeblich den Einsatz neuronaler Netzwerke für den Ausgleich vorhandener Parameterunsicherheiten. Ergebnisse der Flugeigenschaftsuntersuchung des *P360* im Flugversuch

sowie simulative Abschätzungen der aerodynamischen Parameter sind [128] sowie [118] zu entnehmen. In Abbildung 6.2 sind die im Block Flugzeugdynamik simulierten Zusammenhänge zwischen der Aerodynamik, dem Triebwerkverhalten sowie den dargestellten Differentialgleichungssystemen als Blockschaltbild dargestellt.

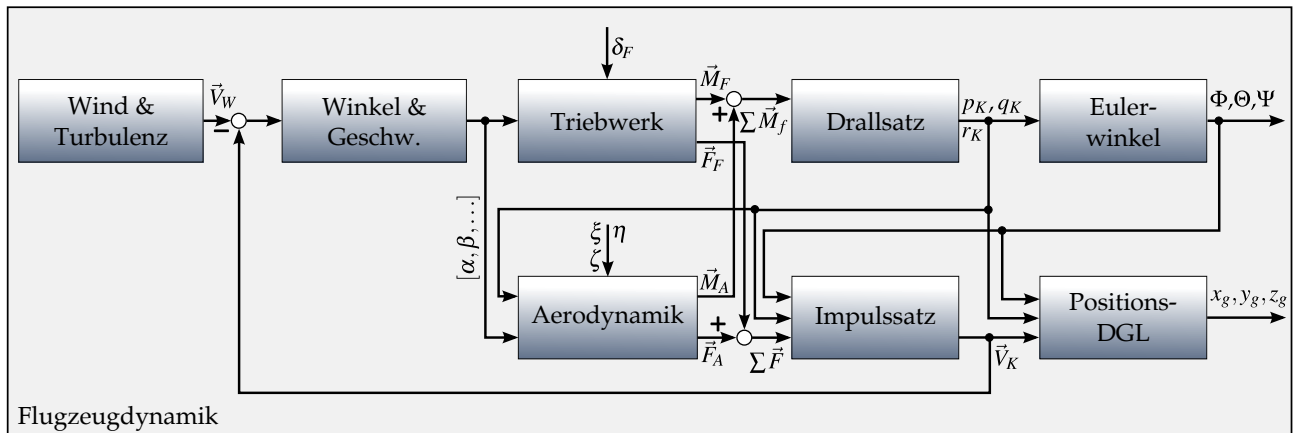


Abbildung 6.2: Blockschaltbild der Flugdynamiksimulation

Neben der Flugzeugdynamik ist, wie in Abbildung 6.1 zu sehen, auch eine Sensormodellierung umgesetzt. Damit soll den in der Realität auftauchenden Effekten wie Messrauschen, Sensordrift und Zeitverzögerung von zumeist günstigen Sensoren kleiner unbemannter Flugsystemen sowie Orientierungsfehlern beim Einbau Rechnung getragen werden. Davon in der Simulation beeinflusst, werden die simulierte Messung der Drehraten und Beschleunigungen, des statischen und des Staudrucks sowie der Positionsbestimmung durch das Satellitennavigationssystem, wie es in [129] beschrieben wird. Ein nach [130] als Quasi-Standard bezeichneter Ansatz für den Ausgleich solcher fehlerbehafteten Messungen ist dabei die Nutzung eines integrierten Navigationssystems. Das zu diesem Zweck in der Simulation verwendete Kalman-Filter in der Tightly-Coupled Closed-Loop-Error-State-Architektur wird in [130, 131] detailliert beschrieben.

Ein weiterer Bestandteil der Simulation ist die Modellierung einer nichtlinearen Aktuatordynamik. Auf diese Weise werden Totzeiten bei der Signalumsetzung sowie Raten- und Ausschlagsbegrenzungen umgesetzt, die ohne entsprechende Berücksichtigung im Regler (vgl. Abschnitt 5.1.5 - PCH) schnell zu Einbußen in der Leistungsfähigkeit bis hin zum Stabilitätsverlust führen können. Aus diesem Grund ist es zweckmäßig, diese schädlichen Effekte bereits in der Simulation zu berücksichtigen, um das ausgelegte Regelungssystem auf die Robustheit gegenüber einer nichtlinearen Aktuatordynamik zu untersuchen. In [132] und [133] sind die dynamischen Eigenschaften von üblicher Weise in kleinen unbemannten Flugsystemen eingesetzten Aktuatoren, wie sie auch in dieser Arbeit Anwendung finden, untersucht und anschließend als Modell umgesetzt worden.

Als letzter der in Abbildung 6.1 dargestellten Blöcke wird in der verwendeten Simulation der Störeinfluss von Wind und Turbulenz berücksichtigt. Dabei wird der konstante Wind-

anteil durch drei Windgeschwindigkeiten in den entsprechenden drei Raumrichtungen als Anteil des Gesamtwindvektors umgesetzt. Schwieriger ist es, die atmosphärische Turbulenz mit ihren zufällig auftretenden Windböen stochastisch korrekt abzubilden. Eine Möglichkeit, dies durch Signale mit ähnlicher Störwirkung zu tun, ist die Verwendung des sogenannten *Dryden*-Turbulenzspektrums [127, 134]. Bei diesem wird durch die Beaufschlagung eines Formfilters mit weißem Rauschen ein Störsignal erzeugt, dessen Leistungsdichtespektrum dem einer im Flug gemessenen Turbulenz ähnelt. Auf diese Weise kann das stochastische Verhalten atmosphärischer Störungen für die Simulation und somit der Auslegung und Untersuchung von Regelungssystemen nutzbar gemacht werden.

6.1.2 Reglerumsetzung

In diesem Abschnitt soll in direkter Anlehnung an [10, 46, 48, 126] in verkürzter Form die Umsetzung des adaptiven Flugreglers auf Basis dynamischer Inversion beschrieben werden. Als Architektur wird dabei eine Kaskade von drei hintereinander geschalteten Regelungsschleifen mit relativem Grad von Eins für jeweils Längs- und Seitendynamik gewählt. Dabei muss auf die Zeitskalentrennung der einzelnen Regelungskaskaden geachtet werden, um mögliche Instabilitäten durch Kopplungseffekte zwischen den unterschiedlichen geregelten Dynamiken zu verhindern. Die Zeitkonstanten der Referenzmodelle für die äußeren und somit langsameren Kaskaden müssen entsprechend deutlich größer als die der nächst inneren Kaskaden sein. Dabei reduziert sich die Gesamtübertragungsbandbreite des Reglers im Vergleich zu einer einzelnen Kaskade mit höherem relativen Grad. In [46] werden ausführlich Vor- und Nachteile des hier verwendeten Ansatzes gegenüber einer Umsetzung mit weniger Schleifen von höherem relativen Grad diskutiert. Ein wesentlicher Aspekt bei der hier gewählten Architektur ist die geringere Komplexität der Fehlerdynamik und die einfachere Erweiterung mit neuronalen Netzwerken.

Das Ziel der vorgestellten Regelung ist die automatische Bahnfolge bei Einhaltung einer konstanten Geschwindigkeitsvorgabe. Dabei kann durch die stetige und somit sprungfreie Verkettung von kubischen Spline-Kurven eine mathematische Beschreibung der Sollbahn erzeugt werden. Dies ermöglicht die Berechnung der seitlichen Bahnabweichung d als globale Qualitätsgröße der Querdynamikregelung. So kann neben der Einhaltung eines vorgegeben Kurses die Ablage von der Sollbahn berechnet und für die Regelung verwendet werden. In einem, der kaskadischen Inversionsregelung vorangestellten, Positionsregler werden zu diesem Zweck aus der aktuellen Position des Flugzeugs in geografischer Länge λ und Breite ϕ sowie dessen Höhe H die benötigten Kommandogrößen berechnet. Dabei handelt es sich um den kommandierten Bahnneigungswinkel γ_k und den aus der Sollbahn berechneten Bahnazimut χ_s , der um die Bahnabweichung d korrigiert wird. Für den kommandierten Bahnazimut mit Kompensation der seitlichen Bahnabweichung wird nach [46] die in Abbildung 6.3 dargestellte Positionsregelung genutzt. Dabei wird die d -Kompensation mit der Geschwindigkeit normiert und das berechnete Bahninkrement $\Delta\chi$ nach oben und unten begrenzt.

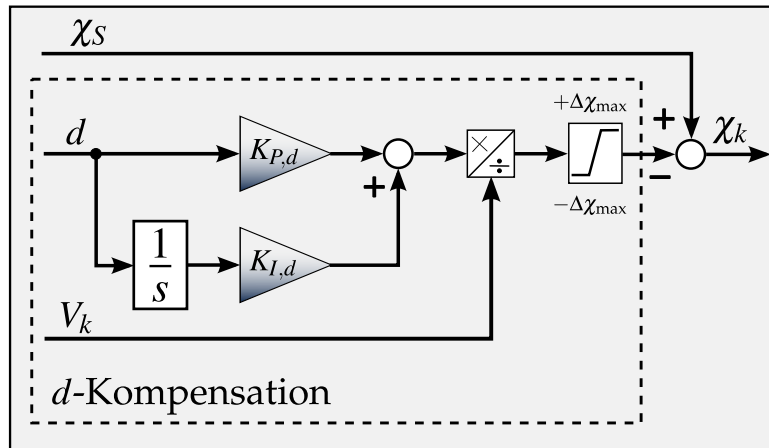


Abbildung 6.3: Kompensation der Bahnablage nach [46]

Für die oben beschriebene Betrachtung einer flachen Erde mit einem als Punktmasse angenommenen Flugzeug und den horizontalen Geschwindigkeitskomponenten \dot{x}_k und \dot{y}_k ergibt sich für den kommandierten Bahnneigungswinkel:

$$\gamma_k = \arctan \frac{\dot{H}_k}{\sqrt{\dot{x}_k^2 + \dot{y}_k^2}}. \quad (6.5)$$

Die auf diese Weise berechneten Kommandogrößen der automatischen Bahnfolgen werden im Anschluss dem Inversionsregler übergeben und stellen somit die ersten Regelgrößen der äußersten Kaskade dar. In Abbildung 6.4 ist ein vereinfachtes Blockschaltbild der kaskadierten Regelungsarchitektur mit vorgeschaltetem Positionsregler dargestellt. Dabei ist die Regelung der konstanten Geschwindigkeitsvorgabe der Übersicht halber nicht mit abgebildet. Wie in der Abbildung zu erkennen ist, wird in dieser Arbeit, wie schon in [10, 48, 90], ausschließlich die Rotationsdynamik und somit die Regelung der körperfesten Drehraten mit neuronalen Netzwerken erweitert. Darüber hinaus können auch die weiteren Kaskaden durch neuronale Netzwerke gestützt werden, wie es beispielsweise in [46] für die Regelung der Bahndynamik erfolgt ist. Mit dem Ziel einer geringeren Systemkomplexität wird an dieser Stelle darauf jedoch verzichtet. Im Folgenden werden die Umsetzungen der einzelnen Schleifen, beginnend bei der innersten Kaskade, in zusammengefasster Form dargestellt.

Regelung der Rotationsdynamik

Ziel der innersten und schnellsten Kaskade ist die Regelung der körperfesten Drehraten $\vec{\omega}_f^{gf} = [p_K, q_K, r_K]^T$. Den Ansatz für die verwendete Inversion liefert dabei der Drallsatz aus Gleichung 6.2. In diesem beeinflusst die Summe aller Momente die am Flugzeug wirkenden Drehbeschleunigungen. Da keine Inversion der Aktuatordynamik vorgenommen wird und somit die berechneten Momente verzögerungsfrei durch die weiter unten dargestellte Steuerflächenallokation in Stellauschläge umgesetzt werden, liegt entsprechend ein direkter Zusammenhang zwischen den Steuergrößen und den ersten Ableitungen der Drehraten vor. Diese Dynamik vom relativen Grad eins kann durch eine näherungsweise Inversion

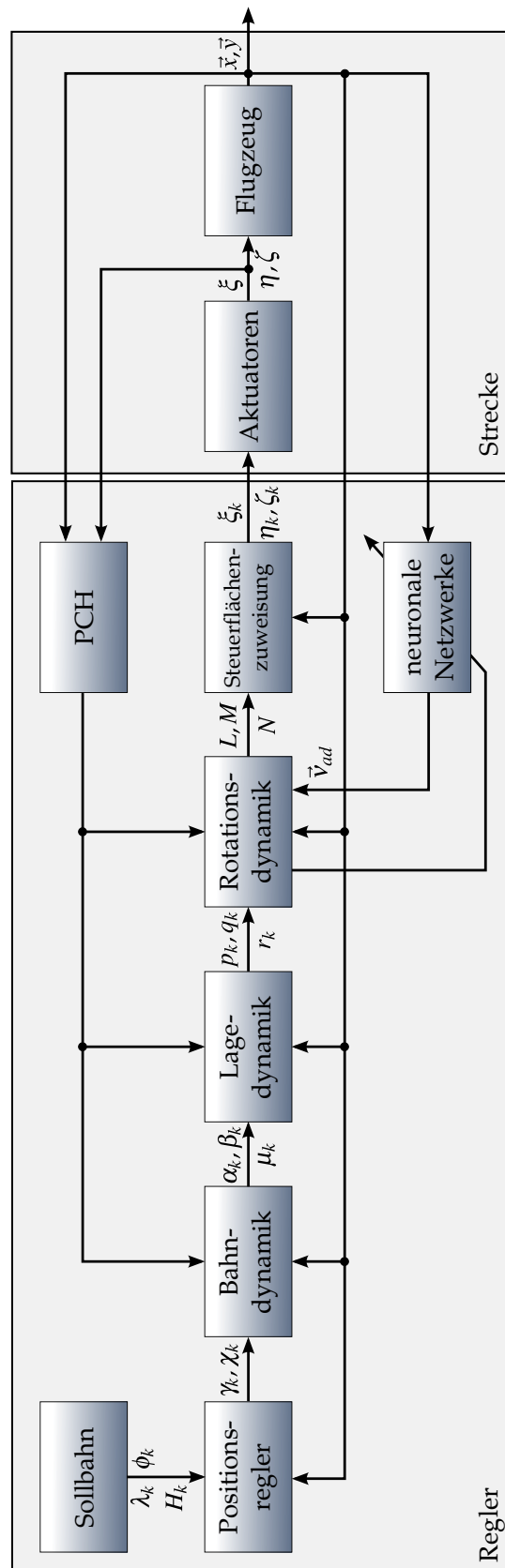


Abbildung 6.4: Vereinfachtes Blockschaltbild des kaskadischen Reglers

in Form von einfachem Auslösen des Drallsatzes nach den Momenten für die Regelung nutzbar gemacht werden. Mit Verwendung des inkrementellen Ansatzes aus [46], bei dem nicht der vollständig benötigte Steuerausschlag berechnet wird, sondern nur das fehlende Inkrement in Ergänzung zum aktuellen Systemzustand, kann die Momentensumme $\sum \vec{M}_f$ aufgeteilt werden in:

$$\sum \vec{M}_f = \underbrace{\left[\vec{M}_0(\vec{x}, \vec{u}) \right]_f}_{\text{aktueller Zustand}} + \underbrace{\left[\delta \vec{M}_k(\delta \vec{u}_k) \right]_f}_{\text{erforderliches Inkrement}} = \begin{bmatrix} L_0 \\ M_0 \\ N_0 \end{bmatrix}_f + \begin{bmatrix} \delta L_k \\ \delta M_k \\ \delta N_k \end{bmatrix}_f. \quad (6.6)$$

Mit Gleichung 6.6 und Auflösen des Drallsatzes nach dem Momenteninkrement ergibt sich für die Inversion der Rotationsdynamik in Abhängigkeit der Pseudosteuergröße $\vec{v}_\omega = \dot{\vec{\omega}}_f^{gf}$:

$$\left(\delta \vec{M}_k \right)_f = \mathbf{T} \cdot \vec{v}_\omega + \vec{\omega}_f^{gf} \times \left(\mathbf{T} \cdot \vec{\omega}_f^{gf} \right) - \left(\vec{M}_0 \right)_f. \quad (6.7)$$

Umgesetzt durch drei Referenzmodelle erster Ordnung folgt für die Brechung der Referenzwinkelbeschleunigungen um die drei Raumachsen zusammengefasst im Vektor $\vec{\omega}_R$ mit Berücksichtigung des Hedging-Signals $\vec{v}_{\omega,h}$:

$$\dot{\vec{\omega}}_R = \frac{1}{T_\omega} \cdot (\vec{\omega}_k - \vec{\omega}_R) - \vec{v}_{\omega,h}. \quad (6.8)$$

Die Pseudosteuergrößen der Rotationsdynamikinversion mit Erweiterung um drei neuronale Netzwerke, für die Beeinflussung je einer Drehachse, ergeben sich schließlich zu:

$$\vec{v}_\omega = \underbrace{\frac{1}{T_\omega} \cdot (\vec{\omega}_k - \vec{\omega}_R)}_{\vec{v}_{\omega,R}} + \underbrace{\mathbf{K}_\omega \cdot (\vec{\omega}_R - \vec{\omega})}_{\vec{v}_{\omega,e}} + \vec{v}_{\omega,ad} + \vec{v}_{\omega,r}. \quad (6.9)$$

Hierbei setzt sich die Matrix \mathbf{K}_ω aus den Verstärkungen der proportionalen Regelungselemente zusammen. Es gilt somit:

$$\mathbf{K}_\omega = \begin{bmatrix} k_p & 0 & 0 \\ 0 & k_q & 0 \\ 0 & 0 & k_r \end{bmatrix}. \quad (6.10)$$

Wie schon in [46] und [48] beschrieben, geht aus der erweiterten Fehlerdynamik

$$\dot{\vec{e}}_\omega + \mathbf{K}_\omega \cdot \vec{e}_\omega = +\vec{\Delta}_\omega - \vec{v}_{ad,\omega} - \vec{v}_{\omega,r} \quad (6.11)$$

mit $\mathbf{A}_{E,\omega} = -\mathbf{K}_\omega$ und $\mathbf{B}_{E,\omega} = \mathbf{E}$ hervor, dass für die Wahl $\mathbf{Q}_{E,\omega} = 2 \cdot \mathbf{E}$ die folgende Lösung für die Ljapunow-Funktion $\mathbf{A}_E^T \cdot \mathbf{P}_E + \mathbf{P}_E \cdot \mathbf{A}_E = -\mathbf{Q}_E$ bestimmt werden kann:

$$\mathbf{P}_\omega = \begin{bmatrix} 1/k_p & 0 & 0 \\ 0 & 1/k_q & 0 \\ 0 & 0 & 1/k_r \end{bmatrix}. \quad (6.12)$$

Daraus resultiert direkt die Zusammensetzung der gefilterte Fehlergröße $\vec{\zeta}_\omega$ als Trainingssignal der drei neuronalen Netzwerke zu:

$$\vec{\zeta}_\omega = \vec{e}_\omega^T \cdot \mathbf{P}_{E,\omega} \cdot \mathbf{B}_{E,\omega} = [e_p/k_p, e_q/k_q, e_r/k_r] . \quad (6.13)$$

Wie bereits in Abschnitt 5.2.2 beschrieben, liegt für das betrachtete System mit relativem Grad von Eins nach Gleichung 6.13 eine direkte Proportionalität des gefilterten Fehlers vom Fehlerzustand \vec{e}_ω der Fehlerdynamik vor. Durch die Anwendung des Gleitzustandsverfahrens kann mit Hilfe der Fehlerzustände des Netzwerks somit ebenfalls die Fehlerdynamik des geschlossenen Regelkreises stabilisiert werden. In Abbildung 6.5 ist die vollständige Umsetzung des um neuronale Netzwerke erweiterten Inversionsreglers für die Drehratenregelung in der Rotationsdynamik dargestellt.

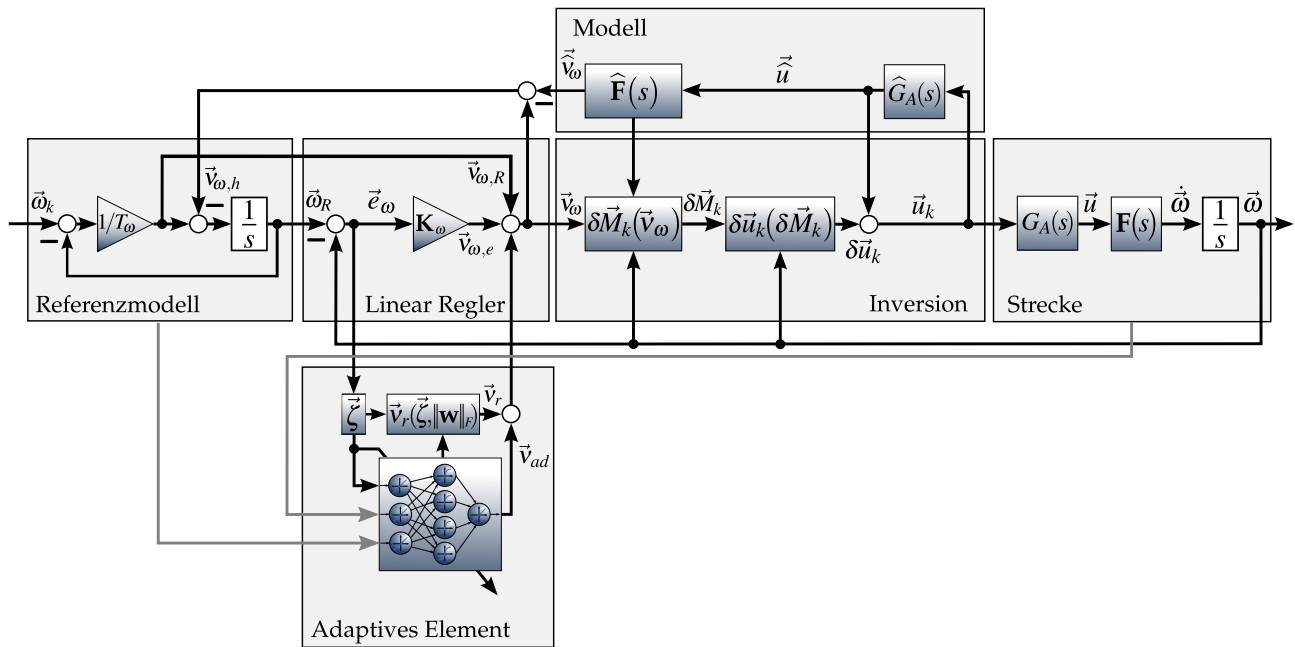


Abbildung 6.5: Umsetzung der Inversionsregelung der Rotationsdynamik nach [46]

Für die Eingangsgrößen der neuronalen Netzwerke kann generell zwischen der Beeinflussung von Längs- oder Seitenbewegung unterschieden werden [48]. So besitzen die beiden neuronalen Netzwerke für die Approximation des Inversionsfehlers in der Roll- bzw. Gierratenregelung nicht zwingend dieselben Eingabegrößen wie das Netzwerk der Nickratenregelung. Wie bereits dargestellt wurde, müssen die Netzeingänge einen ausreichenden Informationsgehalt transportieren, der die Abbildung der jeweiligen geregelten Dynamik und dessen Inversionsfehler erlaubt. Neben den Referenz- und Fehlergrößen handelt es sich dabei also um die wesentlichen und für die Systembeschreibung relevanten Zustandsgrößen. Nach [48] kommen für die Längsbewegung, unter der Annahme der Messbarkeit, vor allem die Nickrate q , der Anstellwinkel α , der Nickwinkel Θ und der Staudruck \bar{q} in Frage. Neben den aktuellen Messsignalen kann zur Approximation des Inversionsfehlers auch die um einen oder mehrere Zeitschritte verzögerte Eingabe einzelner Zustände vorteilhaft sein. Außerdem ist es durch die gezielte Aufnahme von Zuständen

der Seitenbewegung in den Eingabevektor \vec{x} möglich, die natürliche Kopplung von Längs- und Seitendynamik auch im Netzwerktraining zu berücksichtigen. So stellt gerade der Schiebewinkel β als primäre Kopplungsgröße eine wichtige Information dar.

Für das Training der beiden, in der Regelung der Seitenbewegung verschalteten Netzwerke, schlägt [48] die Rollrate p , die Gierrate r , den Schiebewinkel β , den Hängewinkel Φ und erneut den Staudruck \bar{q} vor. Dabei spielt auch die Signalgüte der genutzten Sensordaten eine wichtige Rolle, da stark verrauschte Eingangssignale ebenfalls das generelle Rauschniveau der Netzausgabe erhöhen können.

Regelung der Lagedynamik

In der zweiten Kaskade wird die Regelung der Lagedynamik umgesetzt. Dabei ist es die Aufgabe, aus den Vorgaben der Bahndynamikregelung die benötigten Eingänge der Rotationsdynamik $\vec{\omega}_k$ zu berechnen. Generell ist zwischen der Umsetzung mittels der Eulerwinkel Φ, Θ, Ψ und der körperfesten aerodynamischen Winkel α_K, β_K, μ_K zu unterscheiden [46]. In dieser Arbeit wird, wie auch schon in [10, 48], der zweite Ansatz zur direkten Regelung der aerodynamischen Größen verwendet. Für die Pseudosteuergröße der Lagedynamikinversion folgt:

$$\vec{v}_{Lage} = \begin{bmatrix} v_\alpha \\ v_\beta \\ v_\mu \end{bmatrix} = \begin{bmatrix} \dot{\alpha}_K \\ \dot{\beta}_K \\ \dot{\mu}_K \end{bmatrix} . \quad (6.14)$$

Unter Verwendung der vollständigen Punktmassendifferentialgleichung können die benötigten Ableitungen der aerodynamischen Winkel nach [10] wie folgt geschrieben werden:

$$\dot{\alpha}_K = q_K - \frac{q_A}{\cos \beta_K} - p_K \cdot \cos \alpha_K \cdot \tan \beta_K - r_K \cdot \sin \alpha_K \cdot \tan \beta_K , \quad (6.15)$$

$$\dot{\beta}_K = r_A + p_K \cdot \sin \alpha_K - r_K \cdot \cos \alpha_K , \quad (6.16)$$

$$\dot{\mu}_K = p_A + (q_A \cdot \sin \mu_K + r_K \cdot \cos \mu_K) \cdot \tan \gamma . \quad (6.17)$$

In [10] wird gezeigt, dass die in den Gleichungen auftauchenden Drehraten im aerodynamischen Koordinatensystem durch körperfeste Ausdrücke ersetzt werden können. Dabei müssen jedoch die Vereinfachungen getroffen werden, dass der Schubeinstellwinkel gleich Null ist und der Schiebewinkel durch die Regelung klein gehalten wird, so dass $\sin \beta \approx 0$ und $\cos \beta \approx 1$ gilt. Mit der Aufteilung der am Flugzeug angreifenden Kräfte F_A in den Anteil der aerodynamischen Kräfte W, Q, A und der nur in x_f wirkenden Triebwerkskraft $(F_{TW})_f$ folgt für die Änderung der körperfesten aerodynamischen Winkel:

$$\dot{\alpha}_K = q_K - \frac{A + F_{TW} \cdot \sin \alpha}{m \cdot V_K} + \frac{g}{V_K} \cdot \cos \gamma \cdot \cos \mu_K , \quad (6.18)$$

$$\dot{\beta}_K = p_K \cdot \sin \alpha_K - r_K \cdot \cos \alpha_K - \frac{Q}{m \cdot V_K} + \frac{g}{V_K} \cdot \cos \gamma \cdot \sin \mu_K, \quad (6.19)$$

$$\dot{\mu}_K = p_K \cdot \cos \alpha_K + p_K \cdot \sin \alpha_K + \frac{\tan \gamma}{m \cdot V_K} \cdot \left[(A + F_{TW} \cdot \sin \alpha) \cdot \sin \mu_K - Q \cdot \cos \mu_K \right]. \quad (6.20)$$

Für die Inversion der Lagedynamik können diese drei Gleichungen nun nach der kommandierten und körperfesten Drehrate um die jeweilige Flugzeugachse aufgelöst werden. Dabei werden die nicht messbaren Auftriebs-, Seiten-, und Schubkräfte im parallellaufenden PCH-Modell berechnet. Für die Ausgabe der mittleren Kaskade und somit den Kommandogrößen der innersten Kaskade folgt schließlich:

$$p_{K,k} = v_\mu \cdot \cos \alpha_{K,k} + v_\beta \cdot \sin \alpha_{K,k} + \frac{Q \cdot \sin \alpha_{K,k}}{m \cdot V_{K,k}} - \frac{g}{V_{K,k}} \cdot \cos \gamma_k \cdot \sin \mu_{K,k} \cdot \sin \alpha_{K,k} - \frac{\tan \gamma_k \cdot \cos \alpha_{K,k}}{m \cdot V_{K,k}} \left[(A + F_{TW} \cdot \sin \alpha) \cdot \sin \mu_{K,k} + Q \cdot \cos \mu_{K,k} \right], \quad (6.21)$$

$$q_{K,k} = v_\alpha + \frac{1}{m \cdot V_{K,k}} \cdot (A + F_{TW} \cdot \sin \alpha) - \frac{g}{V_{K,k}} \cdot \cos \gamma_k \cdot \cos \mu_{K,k}, \quad (6.22)$$

$$r_{K,k} = v_\mu \cdot \sin \alpha_{K,k} - v_\beta \cdot \cos \alpha_{K,k} + \frac{Q \cdot \cos \alpha_{K,k}}{m \cdot V_{K,k}} + \frac{g}{V_{K,k}} \cdot \cos \gamma_k \cdot \sin \mu_{K,k} \cdot \sin \alpha_{K,k} - \frac{\tan \gamma_k \cdot \cos \alpha_{K,k}}{m \cdot V_{K,k}} \left[(A + F_{TW} \cdot \sin \alpha) \cdot \sin \mu_{K,k} + Q \cdot \cos \mu_{K,k} \right]. \quad (6.23)$$

Für die Erzeugung der Pseudosteuergrößen als Eingänge der dargestellten Inversion werden erneut drei Referenzmodelle erster Ordnung verwendet. Dabei werden auch in der mittleren Kaskade die Referenzmodelle durch das Hedging-Signal $v_{Lage,h}$ soweit verlangsamt, dass sie Signale kommandieren, denen die Strecke folgen kann. Dabei wird die innere Kaskade aus Sicht der Lagedynamik als Aktuatordynamik interpretiert. Für die Pseudosteuergröße der beispielhaft betrachteten Anstellwinkelregelung folgt:

$$v_\alpha = \frac{1}{T_\alpha} \cdot (\alpha_{K,k} - \alpha_{K,R}) + K_{\alpha,P} \cdot (\alpha_{K,R} - \alpha_K) + K_{\alpha,I} \cdot \int (\alpha_{K,R} - \alpha_K) \cdot dt. \quad (6.24)$$

Dabei ist an Gleichung 6.24 zu erkennen, dass für den Ausgleich des stationären Regelfehlers ein integrierender Regelanteil in der Inversion der Lagedynamik mitaufgenommen wurde. Darüber hinaus wird, wie in [46] vorgeschlagen, sowohl $\dot{\alpha}_{K,R}$ als auch der kommandierte Anstellwinkel $\alpha_{K,R}$ selber limitiert. Die Begrenzung des maximal kommandierten Anstellwinkels erfolgt dabei durch eine einfache Limitierung des Referenzmodellausgangs, um zu verhindern, dass es zu einem Auftriebsabrisch nach der Überschreitung von α_{max} kommt. Die Änderungsrate $\dot{\alpha}_{K,R}$ hingegen wird dynamisch abhängig vom Staudruck sowie vom verbleibenden Abstand zu α_{max} begrenzt. Dies ist notwendig, da die Änderung eines schnell zunehmenden Anstellwinkels nicht sprunghaft gestoppt werden kann und somit die Dynamik bei Annäherung an das Maximum bereits eingeschränkt werden muss [46].

Regelung der Bahndynamik

Die äußerste und von der Dynamik langsamste Kaskade der umgesetzten Regelungsarchitektur generiert die Kommandogrößen für die oben beschriebene Inversion der Lagedynamik. Dies erfolgt durch die Regelung der Bahngrößen in Form des Kurswinkels χ , des Flugbahnwinkels γ und der Fluggeschwindigkeit V_K . Die Geschwindigkeitsregelung wird dabei durch die direkte Erzeugung einer Schubhebelstellung $\delta_{F,k}$ realisiert, so dass in diesem Fall die Bahndynamikkaskade direkt ein Steuersignal berechnet. Für die drei Pseudosteuergrößen ergibt sich der Vektor:

$$\vec{v}_{Bahn} = \begin{bmatrix} v_{V_K} \\ v_{\gamma} \\ v_{\chi} \end{bmatrix} = \begin{bmatrix} \dot{V}_K \\ \dot{\gamma} \\ \dot{\chi} \end{bmatrix}. \quad (6.25)$$

Die erneute Verwendung der Punktmassendifferentialgleichung liefert, unter der bereits getroffenen Annahmen kleiner Schiebewinkel und der Eindimensionalität des Schubs, nach [10] folgende Beschreibung der Bahngrößenänderungen:

$$\dot{V}_K = \frac{F_{TW} \cdot \cos \alpha - W}{m} - g \cdot \sin \gamma, \quad (6.26)$$

$$\dot{\gamma} = \frac{1}{m \cdot V_K} \cdot \left[(A + F_{TW} \cdot \sin \alpha) \cdot \cos \mu_k + Q \cdot \sin \mu_k \right] - \frac{g}{V_K} \cdot \cos \gamma, \quad (6.27)$$

$$\dot{\chi} = \frac{1}{m \cdot V_K \cdot \cos \gamma} \cdot \left[(A + F_{TW} \cdot \sin \alpha) \cdot \sin \mu_k - Q \cdot \cos \mu_k \right]. \quad (6.28)$$

Um nun die mathematische Inversion in Form des Auflöserns der Gleichungen nach den gesuchten Kommandogrößen durchzuführen, ist es notwendig, eine weitere Vereinfachung zu treffen. Um eine eindeutige Lösung für die drei Größen zu finden, wird unter der Bedingung eines möglichst schiefbefreien Flugs die Querkraft als Null angenommen. Dies ist möglich, da das Scheinlot des Flugzeugs bei kleinen β zu jeder Zeit in Richtung der flugzeugfesten z-Achse zeigt und somit die Querkraft vernachlässigbar klein wird [10, 46]. Auf diese Weise lässt sich das Gleichungssystem wie folgt umformen und auflösen:

$$F_{TW,k} = \frac{m(v_{V_K} + g \cdot \sin \gamma_k) + W}{\cos \alpha_K}, \quad (6.29)$$

$$A_k = m \cdot \cos \gamma_k \cdot \sqrt{(V_{K,k} \cdot v_{\chi})^2 + \left(\frac{V_{K,k} \cdot v_{\gamma}}{\cos \gamma_k} + g \right)^2} - \left[m(v_{V_K} + g \sin \gamma_k) + W \right] \cdot \tan \alpha_K, \quad (6.30)$$

$$\mu_{K,k} = \arctan \left(\frac{V_{K,k} \cdot v_{\chi} \cdot \cos \gamma_k}{V_{K,k} \cdot v_{\gamma} + g \cdot \cos \gamma_k} \right). \quad (6.31)$$

Der kommandierte Bahnhängewinkel $\mu_{K,k}$ aus Gleichung 6.31 kann im Anschluss direkt als Eingangsgröße in die Lagedynamik übergeben werden. Für die übrigen Bahngrößen müssen noch weitere Umrechnungen durchgeführt werden, um die in der Lagedynamik

benötigten Sollgrößen zu erhalten. So kann aus dem kommandierten Auftrieb A_k über die Inversion des simulierten Kennfeldes $C_{A,k}(\alpha_k)$, mit

$$C_{A,k} = \frac{A_k}{\bar{q} \cdot S} \quad (6.32)$$

als kommandierter Auftriebsbeiwert in Abhängigkeit des Staudrucks \bar{q} und der Bezugsflügelfläche S , der letztendlich kommandierte Anstellwinkel α_k berechnet werden. Für die Einhaltung der Bedingung kleiner Schiebewinkel wird β_k unabhängig von der Regelung in der Bahndynamik konstant zu Null vorgegeben. Die oben bereits angesprochene Schubhebelstellung $\delta_{F,k}$ geht als letzte der in der Bahndynamik erzeugten Größen aus einem für $F_{TW,k}$ ermittelten Schubkennfeld hervor [127].

Analog dem Vorgehen für die Erzeugung der Pseudosteuergrößen in der Kaskade der Lagedynamik wird auch in der Bahndynamik der Linearregler um einen integrierenden Anteil erweitert. Zusammen mit dem Heding-Signal folgt für die beispielhaft gezeigte Pseudosteuergröße der Bahnwinkelregelung:

$$v_\gamma = \left[\frac{1}{T_\gamma} \cdot (\gamma_k - \gamma_R) \right]_{lim} + K_{\gamma,P} \cdot (\gamma_R - \gamma) + K_{\gamma,I} \cdot \int (\gamma_R - \gamma) \cdot dt . \quad (6.33)$$

Die in Gleichung 6.33 mit dem Index *lim* bezeichnete Begrenzung stellt den Schutz gegen zu langsame und damit nahe an der Abrissgrenze liegende Geschwindigkeiten sicher. Sie ist nach [46] so umgesetzt, dass bei der Annäherung an einen solchen langsamen Geschwindigkeitsbereich nur noch kleine positive Bahnwinkeländerungen $\dot{\gamma}_R$ kommandiert werden können. Bei weiterem Geschwindigkeitsverlust können ab einer Minimalgeschwindigkeit nur noch abnickende Bahnwinkeländerungen vorgegeben werden, um auf diese Weise wieder an Fahrt zu gewinnen. Zusätzlich gibt es eine maximale negative Änderungsrate, die ein zu schnelles Abnicken des Flugzeuges verhindert. Über die Begrenzung des Flugbahnwinkels hinaus, ist auch die Kursänderungsrate $\dot{\chi}$ durch einen maximal zulässigen Wert begrenzt. Dieser beruht dabei auf der noch zulässigen strukturellen Belastung des Flugzeugs und bildet sich somit aus dem maximal erlaubten Lastvielfachen in z-Richtung $n_{z,max}$. Für die maximale Kursänderung folgt nach [46]:

$$\dot{\chi}_{max} = \frac{g}{V_K} \cdot \sqrt{\left(\frac{n_{z,max}}{\cos \gamma_K} \right)^2 - 1} . \quad (6.34)$$

Die Eingangssignale der Bahndynamik kommen wie oben bereits beschrieben aus der, dem Inversionsregler vorangestellten, Positionsregelung und werden aus der aktuellen Position in Relation zu einer vorgegebenen Sollbahn berechnet.

Steuerflächenzuweisung

Nachdem nun die verwendete kaskadische Struktur des umgesetzten Inversionsreglers beschrieben ist, soll im Folgenden die Zuweisung der in der Rotationsdynamik berechneten

Momenteninkremente aus Gleichung 6.7 auf die zur Verfügung stehenden Steuerflächen dargestellt werden. Dieses Vorgehen hat gegenüber der festen Rückführung von Reglerausgängen auf Steuerflächen den Vorteil, dass die Steuerflächenallokation auch dynamisch in Abhängigkeit des Systemzustands verändert werden kann. Somit ist es beispielsweise möglich, einen komplexen Allokationsalgorithmus zu implementieren, der bei einer höheren Anzahl von Steuergrößen als kommandierten Momentenänderungen eine optimale Kombination der erzeugten Ausschläge ermöglicht. Alternativ kann zwischen primären und sekundären Steuerflächen zur Beeinflussung eines Moments unterschieden werden, so dass im Falle einer Beschädigung des primären Stellelements das Sekundäre zur Rekonfiguration herangezogen wird. Untersuchungen dazu mit dem hier gezeigten Inversionsregler sind in [119, 120] dargestellt.

Für die Untersuchungen in dieser Arbeit werden ausschließlich der Querruderausschlag ξ , der Höhenruderausschlag η und der Seitenruderausschlag ζ für die Umsetzung der Momenteninkremente aus der Rotationsdynamik genutzt. Für die Allokation der Momentenänderung auf diese Steuerflächen werden zunächst die benötigten Inkremente der dimensionslosen Momentenbeiwerte berechnet:

$$\delta C_l = \frac{\delta L_k}{\bar{q} \cdot S \cdot b/2}, \quad \delta C_m = \frac{\delta M_k}{\bar{q} \cdot S \cdot l_\mu}, \quad \delta C_n = \frac{\delta N_k}{\bar{q} \cdot S \cdot b/2}. \quad (6.35)$$

Aufgeteilt in Längs- und Seitenbewegung lassen sich die Momentenbeiwertsänderungen in Abhängigkeit der benötigten Steuerflächeninkremente unter Verwendung von aerodynamischen Derivativen wie folgt darstellen:

$$\begin{bmatrix} \delta C_l \\ \delta C_n \end{bmatrix} = \begin{bmatrix} \delta C_{l\xi} & C_{l\zeta} \\ \delta C_{n\xi} & C_{n\zeta} \end{bmatrix} \cdot \begin{bmatrix} \delta \xi \\ \delta \zeta \end{bmatrix}, \quad (6.36)$$

$$\delta C_m = C_{m\eta} \cdot \delta \eta. \quad (6.37)$$

Bei der Lösung des Gleichungssystem 6.36 nach den Steuerflächeninkrementen von Quer- und Seitenruder ist darauf zu achten, dass es zu keinen Singularitäten kommt, so dass zu jedem Zeitpunkt eine eindeutige Lösung für das Steuerkommando existiert [46]. Für die letztendlich an die Aktuatorik kommandierten Steuerungssignale folgt unter Verwendung der geschätzten aktuellen Ausschläge ξ_0 , η_0 und ζ_0 :

$$\xi_k = \xi_0 + \delta \xi, \quad \eta_k = \eta_0 + \delta \eta, \quad \zeta_k = \zeta_0 + \delta \zeta. \quad (6.38)$$

Damit sind alle Elemente des in dieser Arbeit verwendeten Inversionsreglers für die Anwendung auf ein kleines unbemanntes Flugsystem beschrieben. Bei der Parametrisierung der Referenzmodelle sowie der einzelnen Verstärkungsfaktoren der Linearregler wurde entsprechend der Methode zur Bandbreitenoptimierung aus [46] vorgegangen. Dabei wird über die Aufstellung der Übertragungsfunktionen der einzelnen Regelungskaskaden das Hurwitz-Kriterium angewendet, um in Abhängigkeit der Aktuatordynamik Zwangsbedingungen für die Parameter des Reglers zu bestimmen. Mittels einer Kostenfunktion wird

unter Einhaltung der aufgestellten Stabilitätsbedingungen ein Simplex-Algorithmus verwendet, um einen bandbreitenoptimierten Parametersatz für sämtliche Zeitkonstanten und Verstärkungsfaktoren zu erzeugen. Diese Parameter werden in der nichtlinearen Simulation weiter untersucht und wo nötig angepasst, sodass die numerische Vorauslegung nur einen Startpunkt für die finale Parameterauswahl darstellt. Insbesondere die Verstärkungen der Positionsregelung sind in dem beschriebenen Ansatz nicht berücksichtigt und müssen demnach in der Simulation angepasst und parametrisiert werden. Der Nachweis der Anwendbarkeit dieser Auslegungsmethodik für den hier dargestellten Inversionsregler eines kleinen unbemannten Flugsystems sowie eine detaillierte Beschreibung des verwendeten Optimierungsverfahrens sind in [10, 126] gegeben.

6.2 Umsetzung als Fahrzeugregler

Bei der zweiten in dieser Arbeit vorgestellten Anwendung eines Inversionsreglers handelt es sich um die automatische Querführung eines Kraftfahrzeuges. Dabei wird ausschließlich der Kurs und die seitliche Bahnablage durch Ansteuerung des Lenkwinkels an den Vorderrädern mittels dynamischer Inversion geregelt. Im Zuge der Untersuchungen ist ebenfalls ein Inversionsregler für die Längsdynamik entwickelt worden, allerdings konnte dieser nicht im praktischen Fahrversuch getestet werden. Für die direkte Übertragung der simulierten Architektur auf die Anwendung im Fahrversuch wird an dieser Stelle somit ausschließlich der Querdynamikregler vorgestellt.

6.2.1 Simulationsumgebung

Für die simulative Untersuchung des Fahrzeugreglers wird ebenfalls ein geschlossenes System bestehend aus dem Inversionsregler, der Aktuatorik, der Fahrdynamiksimulation, der Sensormodellierung und einem Beobachter zur Aufbereitung der verrauschten Beschleunigungs- und Drehratensignale verwendet. Dabei wird auf die Simulation von Seitenwind und Fahrbahnunebenheiten verzichtet, da die Untersuchung der Tauglichkeit des Reglers zum Ausgleich dieser externen Störungen im Fahrversuch naturgemäß erfolgt. Änderungen am Regelungssystem können durch eine präzise Definition der Schnittstellen zwischen den Eingangs-/Ausgangsgrößen und der eigentlichen Regelung mit sehr geringem Aufwand sowohl simulativ als auch im tatsächlichen Fahrzeug überprüft werden.

Für die Simulation der Fahrdynamik wird in dieser Arbeit ein vollständiges und nicht-lineares Zwei-Spur-Modell in Kombination mit einem nichtlinearen Pacejka-Reifenmodell zur Untersuchung der kinematischen und kinetischen Gegenebenheit verwendet [135–137]. Anders als im oftmals verwendeten Ein-Spur-Modell, bei dem die angreifenden Kräfte beider Reifen je einer Achse zusammengefasst und auf einen in der Fahrbahnebene liegenden Schwerpunkt bezogen werden [138–141], berücksichtigt das Zwei-Spur-Modell neben der unabhängigen Betrachtung der vier Reifen zusätzlich den Fahrzeugaufbau. So wird bei-

spielsweise die Schwerpunktverschiebung durch die sogenannten Wank- (κ) und Nickbewegungen (φ) des Fahrzeugaufbaus berücksichtigt. Diese zusätzlichen rotatorischen Freiheitsgrade führen im Gegensatz zu einer Punktmassenbeschreibung unter anderem zu zusätzlichen Lenkungseinflüssen wie dem Wanklenken, die bei der Verwendung eines Ein-Spur-Modells unberücksichtigt bleiben. Darüber hinaus werden die Aufstandskräfte an allen vier Reifen über eine Feder-Dämpfer-Kombination simuliert, so dass sich die auf die Achsen wirkenden Reifenkräfte neben der Beeinflussung durch den Längsschlupf und den Schräglaufwinkel zusätzlich durch die simulierte Hubbewegung des Fahrzeugs ändern. In Abbildung 6.6 sind zur Verdeutlichung die einzelnen Teilmodule der Fahrdynamiksimulation und der daraus resultierende Signalfluss dargestellt.

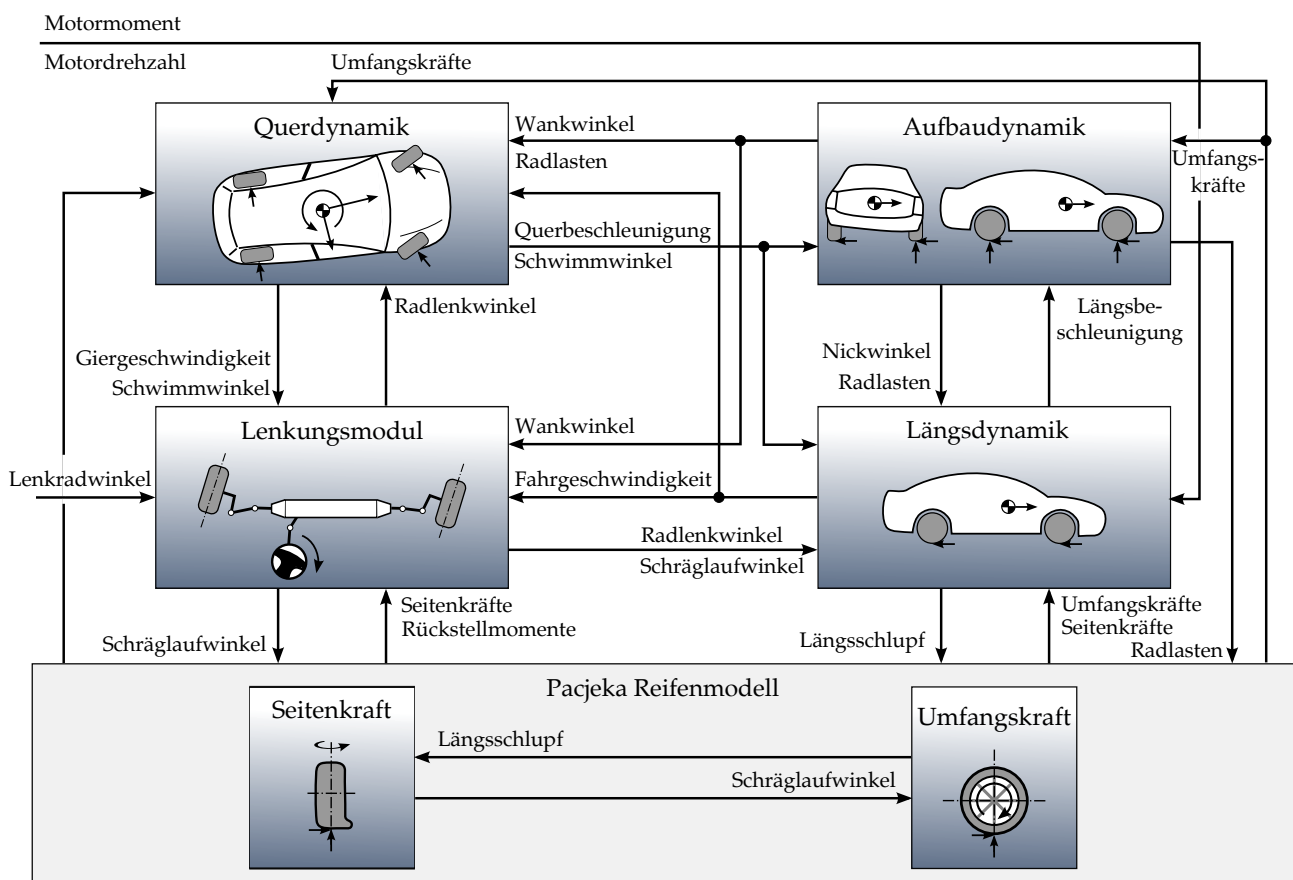


Abbildung 6.6: Signalfluss der Fahrdynamiksimulation direkt nach [135]

Wie schon für die Simulation der Flugzeugdynamik werden auch für die Fahrdynamik die Differentialgleichungssysteme von Drall- und Impulssatz für die Momenten- und Kräftegleichgewichte verwendet. Dabei kann wahlweise der Fahrzeugaufbau von den Achsen freigeschnitten und somit die insgesamt sechs rotatorischen bzw. translativen Freiheitsgrade auf die Systeme Fahrzeugaufbau und Achsenverbund aufgeteilt werden. Bei diesem Ansatz ist für den Aufbau beispielsweise nur noch die Berechnung des Impulssatzes (vgl. 6.1) in Vertikalrichtung z nötig, um auf diese Weise den Hub des Aufbaus zu beschreiben, ohne dabei die x und y Komponente des Achsverbundes zu berücksichtigen. In der in dieser Arbeit verwendeten Fahrdynamiksimulation werden Drall- und Impulssatz jeweils für den

Schwerpunkt des Gesamtfahrzeugs berechnet, so dass beispielsweise ein wankender Aufbau ebenfalls einen Einfluss auf das Giermoment um die Hochachse des Fahrzeuges hat. So gilt für den Drallsatz der Gierwinkelbeschleunigung als Grundlage für die im nächsten Abschnitt beschriebene Inversion der Rotationsdynamik:

$$\ddot{\psi}_f^{gf} = \mathbf{T}^{-1} \cdot \left(\vec{M}_z + \vec{M}_{F_x} + \vec{M}_{F_y} + \vec{M}_{\ddot{x}} + \vec{M}_{\ddot{y}} \right). \quad (6.39)$$

Dabei handelt es sich bei \vec{M}_z um die Rückstellmomente der einzelnen Räder aufgrund des Aufbaus eines Schräglaufwinkels, bei \vec{M}_{F_x} um die Momente als Folge der Umfangskräfte der Reifen und bei \vec{M}_{F_y} um die durch das Einlenken erzeugten Momente nach Aufbau der Reifenseitenkräfte. Die Spurweite als Hebelarm der Reifenumfangskräfte und die Schwerpunktsvor- bzw. Schwerpunktsrücklage als Hebelarm der Reifenseitenkräfte sind dabei dynamisch von der Aufbauauslenkung abhängig. Bei den beiden letzten in Gleichung 6.39 auftauchenden Anteilen, handelt es sich um Momentenanteile infolge der stationären oder beschleunigten Fahrt bei anliegendem Schwimmwinkel β . Eine detaillierte Beschreibung eines Zwei-Spur-Modells mit Darstellung eines Lenkungsmodells und reibwertabhängigem Reifenmodells findet sich unter anderem in [135].

Die Bahnvorgabe erfolgt auch für den Fall der Fahrzeugsimulation durch die Verknüpfung einzelner Bahnelemente zu einem kontinuierlichen und auf diese Weise mathematisch beschreibbaren Kurs. Somit stehen auch für das Szenario des automatischen Fahrens für die Querdynamikregelung wieder die Größen Sollkurs χ_s und seitliche Bahnablage d zur Verfügung. Eine Besonderheit bei der Bahnvorgabe des Fahrzeugreglers ist die Bahngeometrie in Form eines geschlossenen Rundkurses. Auf diese Weise können mehrfache aufeinanderfolgende Runden der gleichen Bahnvorgabe gefahren werden und somit rundenweise die Ergebnisse miteinander verglichen werden. Dabei existiert eine Vielzahl von in der Realität eingefahrenen Bahnvorgaben, aus denen für diese Arbeit einige zur Darstellung der Ergebnisse ausgewählt werden.

6.2.2 Reglerumsetzung

Die im Anschluss gezeigte Umsetzung des Fahrdynamikreglers orientiert sich im wesentlichen Aufbau an der Architektur des Flugreglers aus Abschnitt 6.1.2. So wird erneut eine Aufteilung in mehrere Kaskaden vom relativen Grad eins und die Inversion des Drallsatzes mit anschließender Steuerflächenallokation genutzt. Ebenfalls wird die Rotationsdynamikkaskade, genauer die Regelung der körperfesten Gierrate r_K , mit einem gleitzustandstrainierten neuronalen Netzwerk erweitert. Unterschiedlich zum Flugregler sind dabei die Berechnung nur einer Steuergröße in Form des Lenkwinkels δ und die Verwendung nur zweier anstatt drei Kaskaden. Für die Anwendung des automatischen Fahrens fällt die Lagedynamik weg, so dass in der Bahndynamikkaskade durch die Regelung von $\chi_k \approx \psi_k$ direkt die Eingabe der Rotationsdynamik $\psi_{K,k}$ erzeugt wird.

Auch für die Fahrzeugregelung ist ein der Inversionsregelung vorangestellter Positionsregler für die Kompensation der seitlichen Bahnablage d umgesetzt. Dabei entspricht der Aufbau dem aus Abbildung 6.3 für die d -Kompensation des Flugreglers.

Regelung der Rotationsdynamik

Wie oben erwähnt, ist das Ziel der Rotationsdynamik die Regelung der körperfesten Drehrate um die Hochachse des Fahrzeuges. Die Pseudosteuergröße der inneren Kaskade ergibt sich folglich zu:

$$v_r = \ddot{\psi}_K = \dot{r}_K . \quad (6.40)$$

Nach dem in Gleichung 6.7 dargestellten Ansatz der Drallsatzinversion zur Bestimmung des für die Umsetzung der Pseudosteuergröße benötigten Moments folgt für die Inversion der Rotationsdynamik:

$$M_{k,F_y} = \mathbf{T} \cdot v_r - M_{F_y,3} - M_{F_y,4} . \quad (6.41)$$

Für sowohl die Simulation als auch den Fahrversuch werden dabei ausschließlich die Momente als Folge von Reifenseitenkräften für die Regelung der Rotationsdynamik berücksichtigt. Zur Isolierung der lenkbaren Vorderräder als Haupteinflussgrößen der seitenkraftabhängigen Momentenerzeugung werden die Momente an den beiden hinteren Rädern $M_{y,34}$ abgeschätzt und für die Inversion aus Gleichung 6.41 verwendet. Die darüber hinaus fehlenden Momentenanteile aus Gleichung 6.39 tragen entsprechend zu dem durch das neuronale Netzwerk auszugleichenden Inversionsfehler bei. Für das Referenzmodell erster Ordnung folgt für die Referenzwinkelgeschwindigkeit:

$$\dot{r}_R = \frac{1}{T_r} \cdot (r_k - r_R) - v_{r,h} . \quad (6.42)$$

Die in die Inversion eingehende Pseudosteuergröße setzt sich für die Rotationsdynamik mit Erweiterung um ein neuronales Netzwerk schließlich wie folgt zusammen:

$$v_r = \underbrace{\frac{1}{T_r} \cdot (r_k - r_R)}_{v_{r,R}} + \underbrace{k_r \cdot (r_R - r)}_{v_{r,e}} + v_{r,ad} + v_{r,r} . \quad (6.43)$$

Für die gefilterte Fehlergröße als Trainingssignal für das neuronale Netzwerk kann nach dem Ansatz aus Gleichung 6.13

$$\zeta_r = e_r \cdot \mathbf{P}_{E,r} \cdot \mathbf{B}_{E,r} = \frac{e_r}{k_r} \quad (6.44)$$

geschrieben werden. Zusammengefasst ist die in Abbildung 6.5 dargestellte Struktur für den Fall mit nur einer Regelgröße auch für die Rotationsdynamik der Fahrzeugregelung gültig. Als Eingänge für das neuronale Netzwerk werden die Referenzgierrate r_R , die Referenzpseudosteuergröße $v_{r,R}$, die Fahrzeuggeschwindigkeit V_K , die gefilterte Fehlergröße ζ_r , der Lenkradwinkel δ_l , die Querbefleunigung a_y sowie die Gierrate r_K verwendet.

Eine Besonderheit bei der Anwendung des Fahrdynamikreglers im direkten Vergleich zum Flugregler ist die deutlich geringere Toleranz gegenüber seitlichen Bahnabweichungen. Ist für den Betrieb eines kleinen unbemannten Flugsystems die Einhaltung einer Anstellwinkelbegrenzung oder eines maximalen Lastvielfachen von kritischer Wichtigkeit, so ist für die automatische Fahrt die seitliche Bahnabweichung die wesentliche und stets zu minimierende Größe. So ist leicht verständlich, dass eine zu große seitliche Abweichung schnell zu einem katastrophalen Schaden führen kann. Aus diesem Grund wird in der vorgestellte Positionsregelung die Kompensation der Abweichung d für das automatische Fahren deutlich höher parametrisiert als es für den Flugregler der Fall ist. Dadurch tendieren der Regler und folglich die neuronalen Netzwerke ohne weitere Anpassung jedoch dazu, die Bahnabweichung selbst im Bereich des Messrauschens von wenigen Zentimetern noch weiter zu Null zu regeln. Dieses Verhalten führt dabei zu hochfrequenten Lenkradbewegungen, die zum einen die Aktuatorik verstärkt beanspruchen und zum anderen den Fahrkomfort herabsetzen.

Um dieses Verhalten zu reduzieren, wird der Fehler der Rotationsdynamik e_r und somit auch die gefilterte Fehlergröße ζ_r in Abhängigkeit der seitlichen Bahnablage skaliert. Verwendet wird dabei ein Tangens Hyperbolicus, der ab einer Abweichung von weniger als $\pm 0,2$ Metern den durch die Regelung und das Netzwerk „gesehenen“ Fehler mit dem Faktor σ_d verkleinert. Auf diese Weise wird die Stell- und Lernaktivität in der unmittelbaren Nähe zur Sollbahn reduziert und somit ein parametrierbarer Grenzbereich um die Sollbahnmitte $d = 0$ erzeugt. In Abbildung 6.7 ist die Aufschaltung der e_r -Skalierung für die Verkleinerung der Fehlergröße im Grenzbereich um die Sollbahn dargestellt.

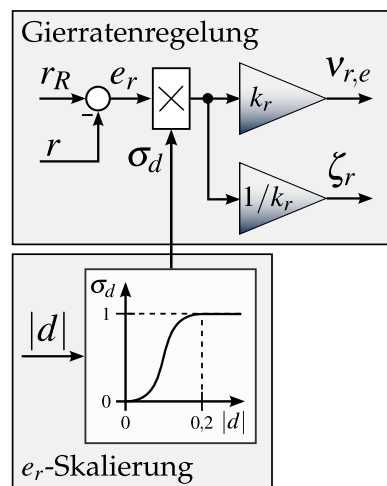


Abbildung 6.7: Skalierung der Fehlergröße e_r in Abhängigkeit der seitlichen Bahnablage d

Regelung der Bahndynamik

Für den hier dargestellten Ansatz eines Querdynamikreglers mit zwei Kaskaden des relativen Grads von Eins übernimmt die Bahndynamik die Aufgabe, aus der Bahnvorgabe χ_k den

Eingang der oben eingeführten Rotationsdynamik zu berechnen. Wie aus der Definition der Pseudosteuergröße

$$v_\chi = \dot{\chi} \approx r \quad (6.45)$$

sofort deutlich wird, handelt es sich dabei bereits um die gesuchte Gierrate als Regelgröße der inneren Kaskade. Mit anderen Worten bedarf es nach der Bildung des Referenzmodells

$$\dot{\chi}_R = \frac{1}{T_\chi} \cdot (\chi_k - \chi_R) - v_{\chi,h} \quad (6.46)$$

und der Zusammensetzung der Pseudosteuergröße zu

$$v_\chi = \frac{1}{T_\chi} \cdot (\chi_k - \chi_R) + K_{\chi,P} \cdot (\chi_R - \chi) + K_{\chi,I} \cdot \int (\chi_R - \chi) \cdot dt \quad (6.47)$$

keiner weiteren Inversion, so dass $v_\chi \approx r_k$ aus Gleichung 6.47 direkt als Eingangsgröße der Rotationsdynamik hervorgeht.

Steuergrößenzuweisung

Für die Umsetzung der Steuergrößenzuweisung sind, aufgrund der unterschiedlichen in Gleichung 6.39 sichtbaren und vom Lenkwinkel δ abhängigen Momentenanteile, verschiedene Ansätze realisierbar. Wie in Abbildung 6.8 dargestellt ist, bildet die Schwerpunktrücklage l_V den primären Hebelarm für die an den Reifen wirkenden Seitenkräfte F_y . Je größer allerdings der Lenkwinkel δ wird, desto mehr Anteil hat die Umfangskraft F_x über diesen Hebelarm auf das Gesamtgierrmoment. Somit können unter Verwendung einer vollständigen Reifenkraftschätzung sowohl F_x als auch F_y für die Berechnung des benötigten Lenkwinkels δ verwendet werden.

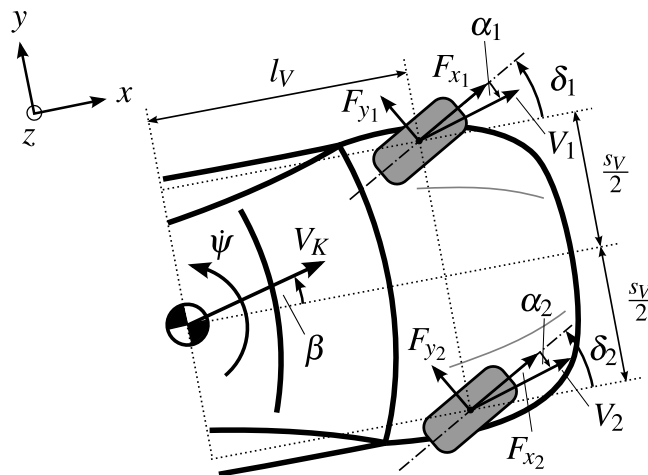


Abbildung 6.8: Kräfte und Winkel an der Vorderachse des Fahrzeugs

Wie bei der Darstellung der Rotationsdynamik bereits erwähnt, wird in dieser Arbeit jedoch auf die Schätzung der Reifenumfangskräfte für die Verwendung in der Regelung

verzichtet, so dass ausschließlich die durch das Einlenken entstehenden Seitenkräfte für die Erzeugung des kommandierten Giermoments berücksichtigt werden. Auf diese Weise kann der Zusammenhang zwischen dem Lenkwinkel δ und den resultierenden Seitenkräften genutzt werden, um über die Abschätzung des benötigten Schräglaufwinkels das in der Rotationsdynamik berechnete Sollmoment umzusetzen.

Die vorwärtsgerichtete Wirkkette des Seitenkraftaufbaus beginnt mit der Änderung des Lenkradwinkels δ_l , was über die Lenkübersetzung zunächst zu einer Winkeländerung δ der angelenkten Räder führt. Durch diesen Lenkwinkel entsteht eine Differenz zwischen der Orientierung der Rads und der Richtung des Geschwindigkeitsvektors und erzeugt auf diese Weise den sogenannten Schräglaufwinkel α . Über die Reifensteifigkeit und den aktuell vorherrschenden Reibwert führt dieser, unter Vernachlässigung verschiedener weiterer Effekte, primär zum Aufbau einer seitlichen Reifenkraft F_y . In Abhängigkeit des Hebelarms l_V vom Rad zum Fahrzeugschwerpunkt führt diese Kraft schlussendlich zum Giermomentenanteil M_{F_y} . In Abbildung 6.9 ist die Steuerflächenzuweisung als Inversion des Giermomentenaufbaus durch eine Lenkradwinkeländerung dargestellt.

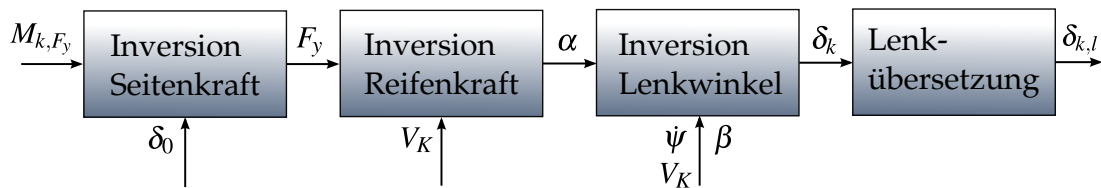


Abbildung 6.9: Aufbau der Steuerallokation zur Lenkwinkelberechnung

Eine besondere Herausforderung stellt bei dieser Art der Steuergrößenzuweisung die Inversion des Reifenkraftaufbaus dar, da dieser von zahlreichen dynamischen und schwer messbaren Faktoren wie beispielsweise den Radaufstandskräften F_z , dem Radschlupf S und dem Radsturz γ abhängt. Aus diesem Grund wird in Einklang mit dem Verzicht auf die Reifenumfangskraftschätzung für die Umsetzung des Fahrdynamikreglers ausschließlich eine lineare Näherung des Reifenkraftaufbaus in Abhängigkeit der aktuellen Geschwindigkeit V_K verwendet. Dabei wird für das lineare Verhältnis zwischen Schräglaufwinkel und Reifenkraft eine Konstante Γ angenommen.

Für die Berechnung der benötigten Seitenkraft zur Erzeugung des kommandierten Giermoments M_k folgt beispielhaft für das vordere linke Rad mit dem Hebelarm l_V :

$$F_{y,1} = \frac{M_k}{2 \cdot \cos(\delta_0) \cdot l_V} . \quad (6.48)$$

Wie oben beschrieben, wird die berechnete Seitenkraft $F_{y,1}$ anschließend mit einer Konstanten Γ in Abhängigkeit der aktuellen Fahrzeuggeschwindigkeit V_K zum benötigten Schräglaufwinkel umgerechnet. Es gilt:

$$\alpha_1 = \frac{F_{y,1}}{\Gamma \cdot V_K} . \quad (6.49)$$

Der berechnete Schräglaufwinkel α_1 wird nun, durch die Inversion der geometrischen Beziehungen am Rad, mit s_V als die Spurweite in den kommandierten Lenkwinkel δ_k umgerechnet. Es folgt:

$$\delta_k = \alpha_1 + \arctan \left(\frac{V \cdot \sin(\beta) + \dot{\psi} \cdot l_v}{V \cdot \cos(\beta) - \left(\dot{\psi} \cdot \frac{s_V}{2}\right)} \right). \quad (6.50)$$

Im Anschluss wird dieser Lenkwinkel mit der wirksamen Lenkübersetzung zum kommandierten Lenkradwinkel $\delta_{k,l}$ umgerechnet und auf den Aktuator in Form einer Lenkmaschine gegeben. Für die Wahl der Zeitkonstanten und Reglerverstärkungen kann das identische Vorgehen, wie es in Abschnitt 6.1.2 für den Flugregler beschrieben wurde, umgesetzt werden. Dabei fällt lediglich eine zu parametrierende Kaskade für die Anwendung im Fahrzeug weg. Gerade im Fahrversuch gilt es, die aus der Simulation stammenden Werte noch einmal zu bewerten und wo nötig abzuändern. Auf diese Weise kann auf in der Simulation getroffene und in der Realität nicht mehr zutreffende Annahmen, wie dem Wegfall von Fahrbahnanregungen und Seitenwinden sowie einem Reibwert von Eins, reagiert werden. Im folgenden Kapitel wird eine Auswahl der erzeugten Ergebnisse für sowohl den Flug- als auch den Fahrzeugregler dargestellt.

Kapitel 7

Darstellung ausgewählter Ergebnisse

In diesem Kapitel sollen die entwickelten Gleitzustandslernverfahren in den vorgestellten Regelungsarchitekturen verglichen und auf ihre Leistungsfähigkeit hin untersucht werden. Dabei werden neben Nominalfällen verschiedenartige Fehler- und Schadensfälle betrachtet, um eine Aussage über die Erhöhung der Fehlertoleranz durch den Einsatz gleitzustands-trainierter neuronaler Netzwerke treffen zu können. Die angestellten Betrachtungen und ausgewählten Ergebnisse werden hierbei in drei Abschnitte unterteilt.

Zu Beginn soll als direkte Anknüpfung an die vorangegangenen Arbeiten in [48] der Flugregler als Teil der in Abschnitt 6.1.1 vorgestellten Simulationsumgebung untersucht werden. Für diesen Zweck wird eine beispielhafte 300-sekündige Flugmission mit atmosphärischer Störung durch Wind und Turbulenz simuliert. Neben der Analyse des fehlerfreien Nominalfalls werden zusätzlich der Ausgleich eines künstlich aufgetragenen Inversionsfehlers sowie die dynamische Veränderung der statischen Stabilität $C_{m\alpha}$ untersucht. Im Anschluss wird der Fahrzeugregler für die Querdynamikregelung des in Abschnitt 6.2.1 dargestellten nichtlinearen 2-Spur-Modells auf einem Rundkurs untersucht. Dabei liegt der Fokus bei der einstündigen Simulation des Nominalfalls auf dem Langzeittraining und dem Vergleich des Trainingsergebnisses über mehrere Runden. Für die Bewertung der Fehlertoleranz durch den Einsatz neuronaler Netzwerke wird ein Reifenplatzer am vorderen linken Rad simuliert und die Leistungsfähigkeit der unterschiedlichen Lernverfahren beim Ausgleich dieser kritischen Degradation verglichen. Als abschließende Untersuchung der vorgestellten Verfahren wird der Fahrzeugregler in praktischen Fahrversuchen eingesetzt und dessen Regelgüte im direkten Vergleich mit einem linearen Referenzregler bewertet. Dabei wird als induzierter Fehlerfall die Reaktion auf einen aufgetragenen Lenkradsprung untersucht.

7.1 Flugsimulation

Zunächst sollen die Gleitzustandsverfahren erster und zweiter Ordnung für das Training neuronaler Netzwerke im vorgestellten Flugregler untersucht werden. Zu diesem Zweck wird eine beispielhafte Flugmission, wie sie in Abbildung 7.1 zu sehen ist, mit einer konstan-

ten Geschwindigkeitsvorgabe von 23 m/s simuliert. Dabei markiert das Kreuz im rechten Teil der Abbildung den Startpunkt und der Pfeil die Flugrichtung.

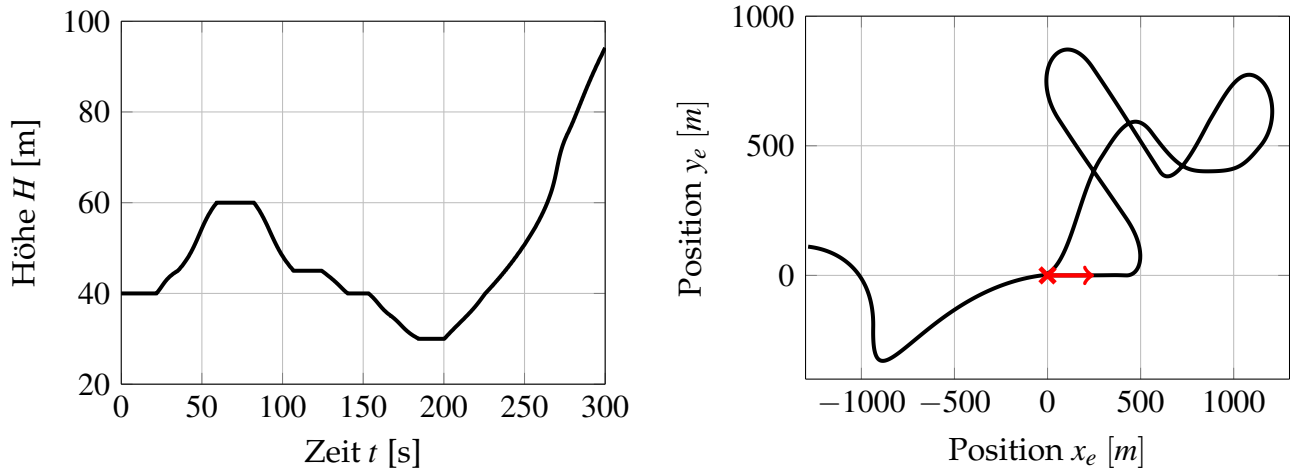


Abbildung 7.1: Bahnvorgabe der Flugsimulation in Längs- und Seitenbewegung

Die dargestellte Kombination von Steig-, Sink- und Kurvenflügen repräsentiert ein möglichst breites Spektrum an unterschiedlichen Flugzuständen mit stellenweise hochdynamischen Übergängen zwischen den einzelnen Teilabschnitten. So wird beispielsweise bei Sekunde 110 zeitgleich zum Abfangen nach einem Sinkflug eine 90° Kurswechsel kommandiert, so dass hohe Drehraten notwendig sind, um dieser Vorgabe folgen zu können.

7.1.1 Nominalfall mit leichten Parameterunsicherheiten

Zu Beginn wird der fehlerfreie Nominalfall mit moderaten Parameterunsicherheiten der Inversionsregelung simuliert. Dabei sind sämtliche für die Inversion angenommenen Flugzeugparameter um 10% erhöht und erzeugen auf diese Weise einen anhaltenden Inversionsfehler. Zusätzlich wird, wie in Abschnitt 6.1.1 beschrieben, ein konstanter Wind von 1 m/s in allen drei Raumrichtungen mit überlagerter Turbulenz simuliert. So kommt es zu Windgeschwindigkeiten mit kurzfristigen Änderungen von bis zu ± 5 m/s. In den Abbildungen 7.2 und 7.3 sind die Soll-Ist-Verläufe der Regelungskaskaden von Längs- und Seitendynamik sowie die seitliche Bahnablage für das Gleitzustandstraining erster Ordnung beispielhaft mit $\tau = 0,7$ und zunächst moderatem $\lambda = 100$ dargestellt. Neben den geringen Abweichung im Höhenverlauf mit einzelnen Überschwingern, wie zum Beispiel nach dem Start der Simulation aufgrund der für das ideal modellierte Flugzeug durchgeführten Trimmung, ist den dargestellten Verläufen klar der Einfluss der atmosphärischen Störungen zu entnehmen. Hervorzuheben ist dabei der gezeigte Verlauf der Nickratenregelung. Trotz des dargestellten sprunghaften Sollgrößenverlaufs ist es der Rotationsdynamikinversion möglich, dieser dynamischen Vorgabe zu folgen. Dank des stabilen Ausgleichs des Inversionsfehlers durch das neuronale Netzwerk, ist dabei keine gleichbleibende Abweichung der Verläufe zu erkennen, wie sie für Parameterunsicherheiten charakteristisch ist.

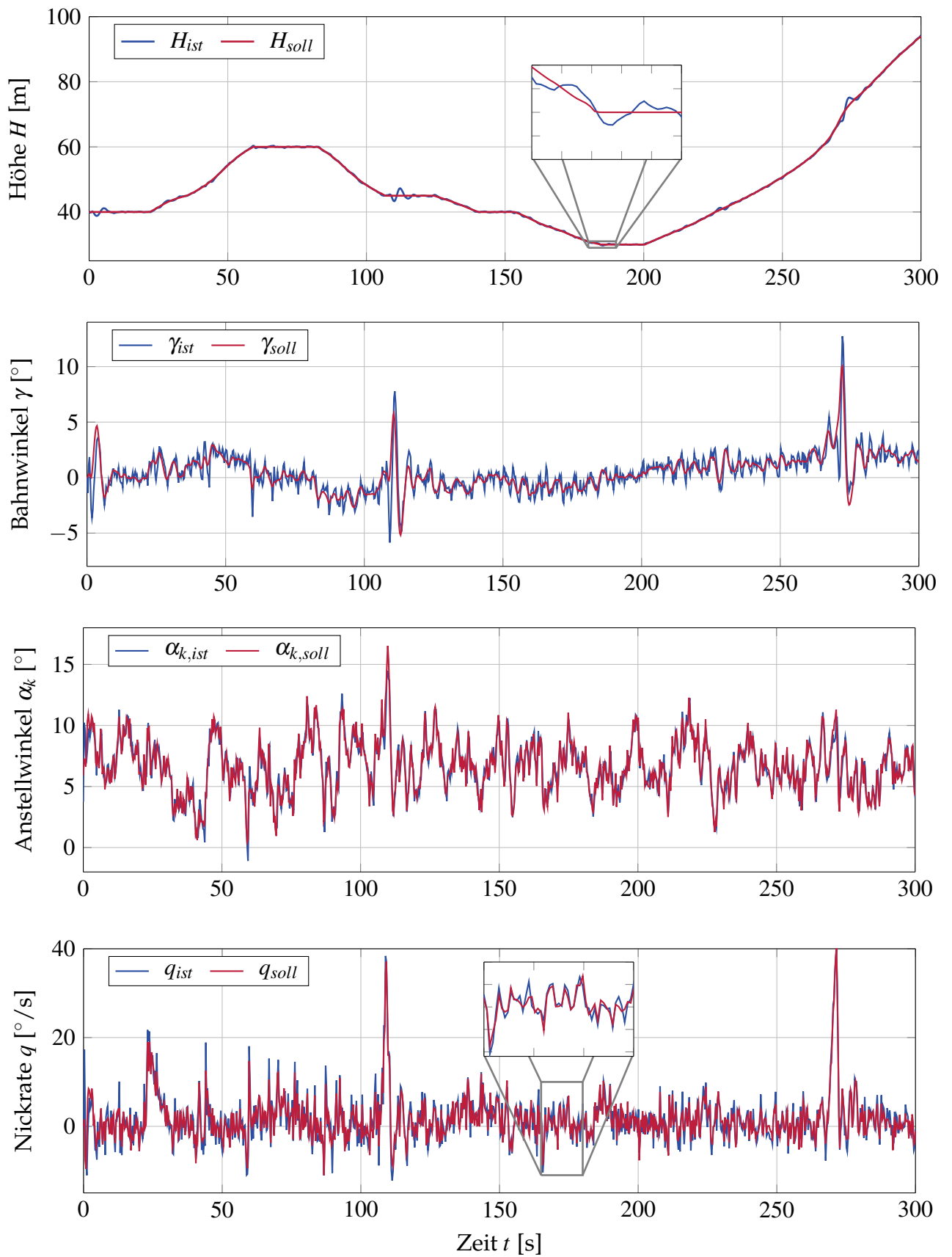


Abbildung 7.2: Größen der Längsdynamikregelung mit leichten Parameterunsicherheiten und Windeinfluss

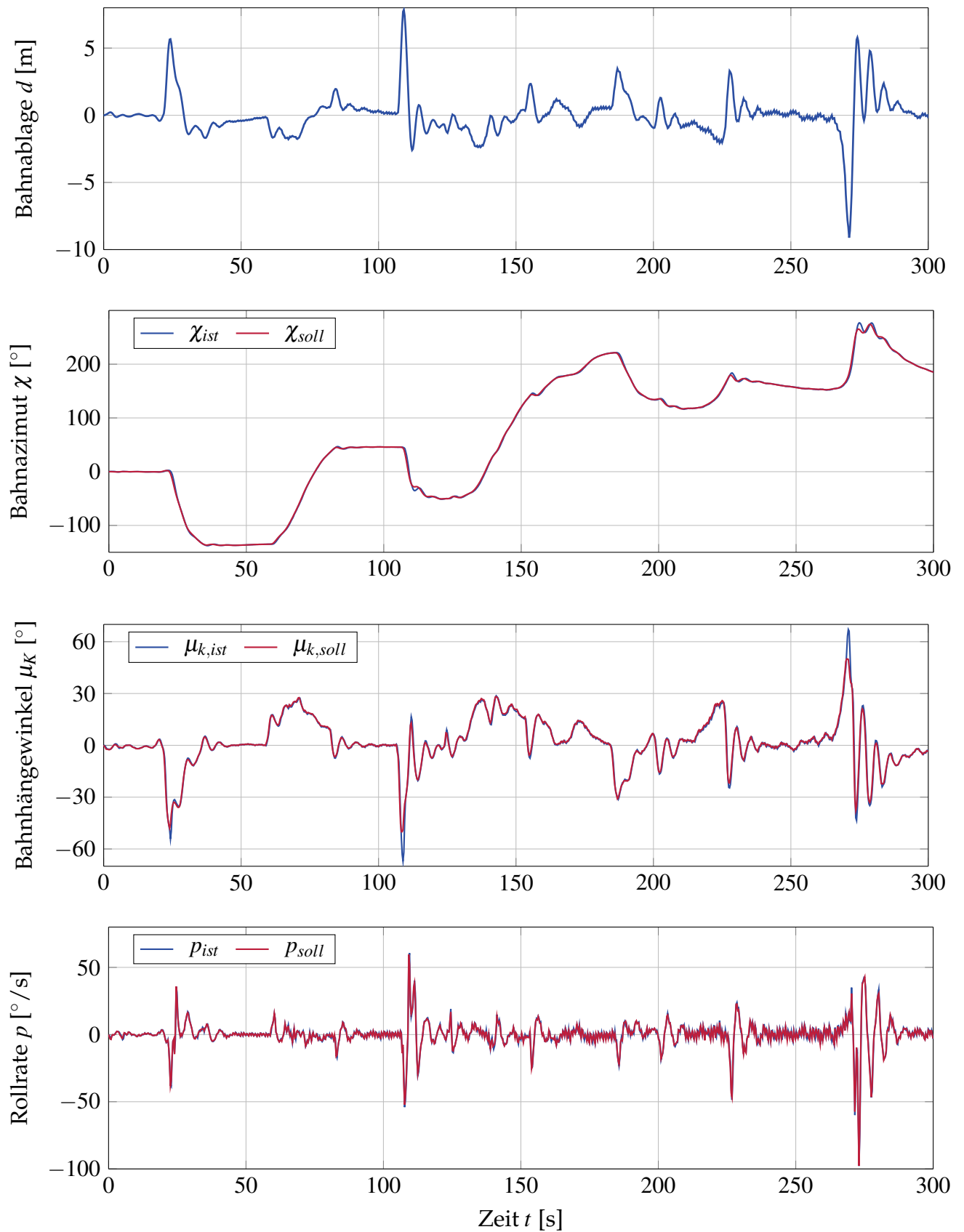


Abbildung 7.3: Größen der Seitendynamikregelung mit leichten Parameterunsicherheiten und Windeinfluss

Auch in den dargestellten Zustandsverläufen der Regelung des Bahnazimuts χ , des Bahnhängewinkels μ_k und der Rollrate p sind nur kleine Abweichungen von der Sollvorgabe zu erkennen. Dabei kommt es wie zu erwarten bei der seitlichen Bahnablage d in den Abschnitten einer Kursänderung zu etwas höheren Abweichungen als für die Geradeausflüge. Insbesondere an den oben erwähnten Bahnübergängen mit einer Kombination aus Kurs- und Höhenänderung kommt es zu Ablagen von mehr als 5 Metern. So führt der Übergang von einer steilen Rechts- in eine Linkskurve kombiniert mit der nochmaligen Steigerung der kommandierten Höhenänderung bei Sekunde 260 zu einer Bahnablage von nahezu -10 Metern, was mit der durch das Flugzeug nicht abbildbaren Sollbahn zu erklären ist. Diese deutliche Bahnablage wird im Anschluss jedoch schnell und stabil abgebaut. Dabei ist wie auch schon für die Längsdynamikregelung eine hohe Güte der Drehratenregelung durch den Einsatz des neuronalen Netzwerks zu beobachten. Des Weiteren sind die dauerhaften, aus den Positions-Updates des Kalman-Filters resultierenden, Sprünge in der Rollratenregelung zu sehen. Durch die jede Sekunde einmal stattfindende Korrektur der Position und folglich der Bahnablage werden durch die d -Kompensation aus Abbildung 6.3 diese Positionssprünge bis zur innersten Kaskade weitergeleitet und führen dort zu der in Abbildung 7.3 für die Rollrate zu sehenden Dynamik.

Insgesamt kann festgehalten werden, dass die vorgestellte Flugregelungsarchitektur für die automatische Bahnfolge mit Sensorrauschen, atmosphärischen Störungen und leichten Modellunsicherheiten unter Verwendung dreier gleitzustandtrainierter Netzwerke mit einer beispielhaften Parametrisierung von $\tau = 0,7$ und $\lambda = 100$ zu einem präzisen und robusten Ergebnis führt. Nach der in diesem Abschnitt dargestellten Leistungsfähigkeit im Nominalfall, soll im folgenden Teil der Ergebnisauswertung die Konvergenzgeschwindigkeit der Gleitzustandsverfahren erster und zweiter Ordnung am Beispiel eines induzierten Inversionsfehlers untersucht werden.

7.1.2 Netzwerkverhalten bei sprunghaftem Inversionsfehler

Neben dem stabilen Ausgleich vorhandener Inversionsfehler als Folge von Modell- und Parameterunsicherheiten wird im Anschluss untersucht, in wie weit die eingesetzten neuronalen Netzwerke zur Kompensation sprunghafter Fehler in der Lage sind. Das maßgebliche Gütekriterium für den Ausgleich kurzfristiger Systemveränderungen ist dabei die Konvergenzgeschwindigkeit der Netzwerke. So ist es das Ziel, durch eine möglichst schnelle Kompensation von auftretenden Systemdegradationen, den Eintritt in kritische Flugzustände zu verhindern und auf diese Weise die Fehlertoleranz des adaptiven Gesamtsystems im Vergleich zu einem reinen robusten System zu erhöhen. Als vergleichbarer Testfall wird die sprunghafte Aufschaltung eines konstanten Fehlersignals von 3 rad/s^2 in der Nickratenregelung untersucht. Im linken Teil der Abbildung 7.4 sind zu diesem Zweck die Reaktionen der Netzausgabe v_{ad} , des robustifizierenden Terms v_r und der Kombination beider Größen $v_{ad,r}$ auf diesen induzierten Inversionsfehler dargestellt. Dabei wird der Fehler als Teil der Pseudosteuergröße v_q aufgeschaltet und ist somit durch das Netzwerk nicht unmittelbar

beobachtbar. Erst durch die eintretende Flugzeugreaktion und den in Folge dessen ansteigenden gefilterten Fehler ζ_q kann das Netzwerk auf die Fehleraufschaltung reagieren. Als Ergebnis der vorangegangenen Untersuchungen in Abschnitt 4.3.3 wird der Schaltfunktionsparameter auf $\lambda = 300$ gesetzt, da für die untersuchten Trainingsszenarien für diesen Wert eine deutliche Reduktion des mittleren quadratischen Fehlers gegenüber $\lambda = 100$ festgestellt wurde.

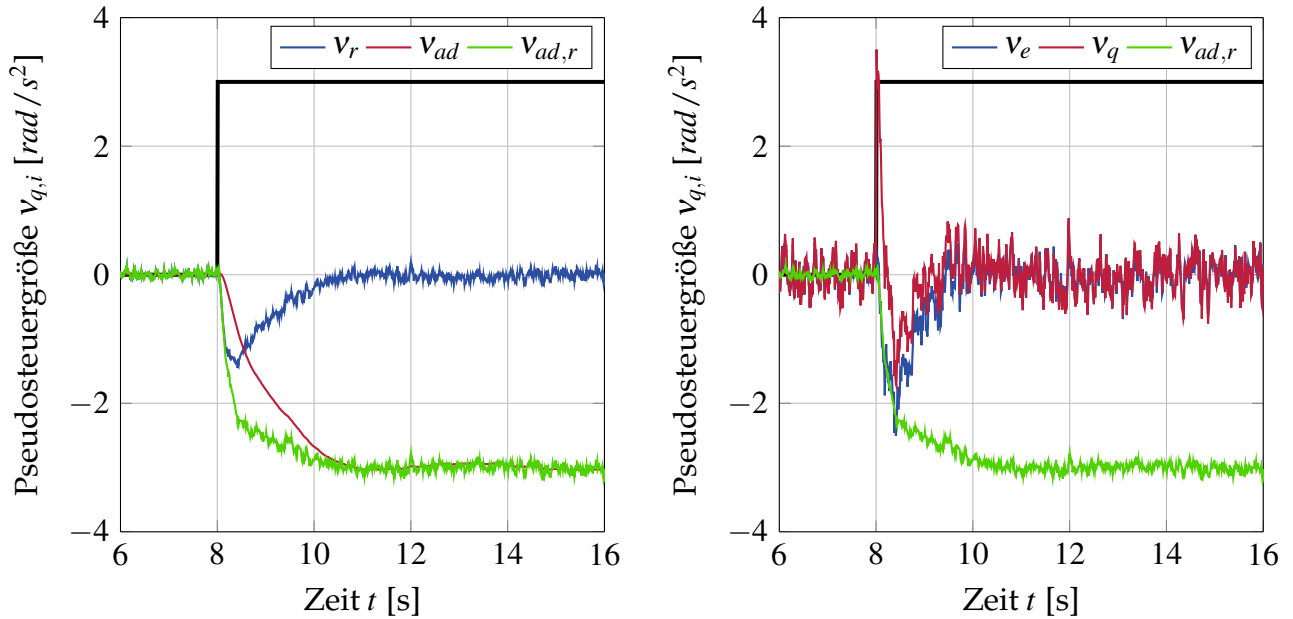


Abbildung 7.4: Reaktion auf sprunghaften Inversionsfehler in der Längsbewegung

Im linken Teil von Abbildung 7.4 ist deutlich die kompensierende Wirkung von sowohl der Netzausgabe als auch des robustifizierenden Terms zu sehen. Als direkte Reaktion auf den Sprung steigt zunächst v_r , um nach beginnender Approximation durch das neuronale Netz wieder gegen Null abzuklingen. Dabei ist klar der Ausgleich des Fehlers durch die sich verändernde Ausgabe des Netzwerks zu erkennen. Die Kombination der beiden Anteile erlaubt somit einen schnellen und stabilen Ausgleich des Fehlers, so dass nach zwei Sekunden der induzierte Inversionsfehler vollständig kompensiert ist. Im rechten Teil der Abbildung ist dies auch anhand der Verläufe der Linearreglerausgabe v_e und der kombinierten Gesamt pseudosteuergröße v_q zu sehen. Bei Aufschaltung des Fehlers nehmen diese zunächst zu, klingen im Anschluss infolge der angepassten Netzwerkausgabe aber schnell ab und erzeugen wieder Ausgaben um die Nulllage herum. Somit ist das Netzwerk in der Lage die Fehlerdynamik nach kurzer Zeit auf einen Stand zurück zu führen, wie er vor der Aufschaltung des Fehlers war.

Für die Untersuchung des Einflusses der Intervallparameter auf das Verfahrens erster (SMC-FO) bzw. zweiter Ordnung (SMC-SO) sind in Abbildung 7.5 die Nickratenfehler e_q beider Trainingsalgorithmen für unterschiedliche τ und τ_s dargestellt. Dabei wird für beide Ansätze $\lambda = 300$ gewählt und das τ für SMC-SO gleichbleibend auf 0,7 gesetzt.

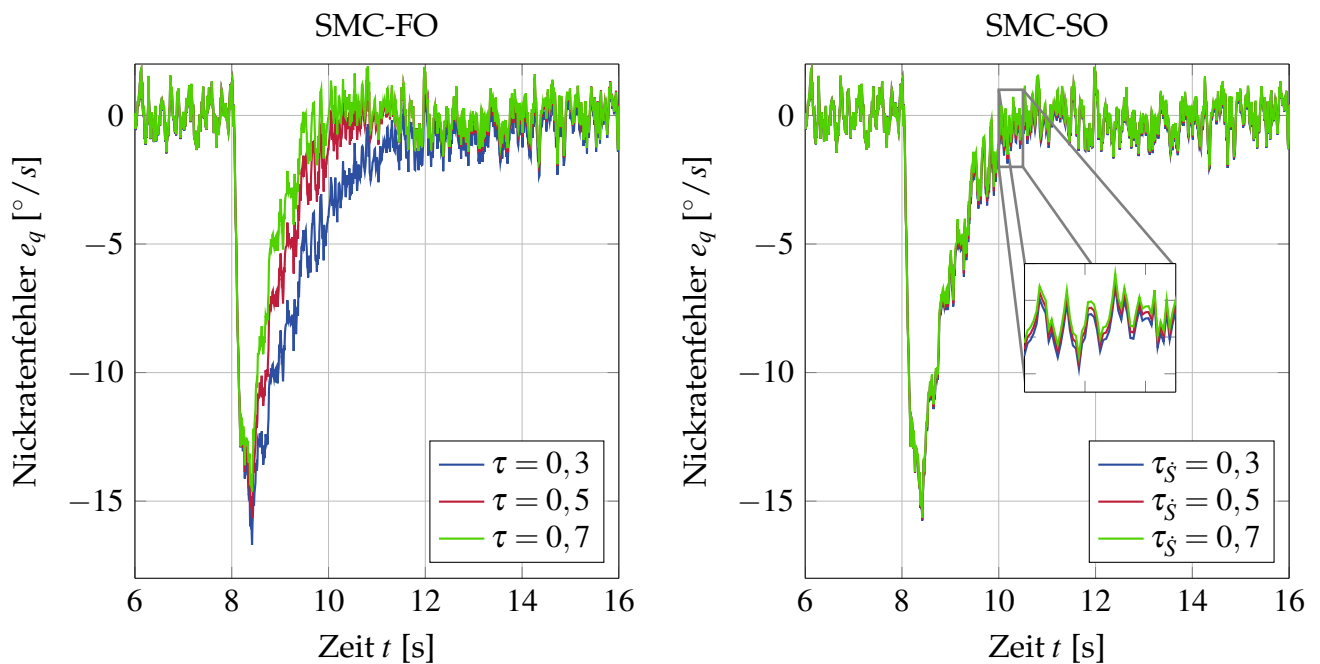


Abbildung 7.5: Ausgleich eines sprunghaften Inversionsfehlers bei unterschiedlichen Intervallparametern

Als direkte Bestätigung der Untersuchungen in Abschnitt 4.3.3, ist im linken Teil der Abbildung die Beschleunigung der Fehlerkompensation durch eine höhere Wahl des Intervallparameters τ zu sehen. Dabei kommt es sowohl zu einer Reduktion des Fehlermaximums als auch zu einem schnelleren Ausgleich. Nach der abgeschlossenen Approximation des Fehlers ist eine starke Ähnlichkeit der drei Verläufe zu erkennen, so dass durch die Wahl von $\tau = 0,7$ ausschließlich die Fehlerreaktion beschleunigt, nicht aber die generelle Netzwerkausgabe verändert wird. Für das Trainingsverfahren zweiter Ordnung ist für alle drei abgebildeten Verläufe ein nahezu identisches Ergebnis zu sehen. So wird erst in der Vergrößerung deutlich, dass auch höhere Werte von τ_S zu einer relativen Verbesserung gegenüber kleineren Intervallfaktoren führen. Dabei kommt es, wie schon in Abschnitt 4.4.3 dargestellt, nur für $\tau_S = 0,7$ zu einer absoluten Verbesserung gegenüber dem Gleitzustandstrainings erster Ordnung, dessen Verlauf der Übersichtlichkeit halber nicht mit abgebildet ist.

Als abschließende Untersuchung des induzierten Fehlerfalls sind Abbildung 7.6 die Lernratenbeträge der in Abbildung 7.5 dargestellten Fehlerverläufe abgebildet. Dabei ist erneut das Trainingsverfahren erster Ordnung auf der linken und das Verfahren zweiter Ordnung auf der rechten Seite dargestellt. Für beide Verfahren ist infolge der Fehleraufschaltung ab Sekunde 8 klar eine Veränderung der Lernratenberechnung zu erkennen. So ist es den Gleitzustandsverfahren durch eine Reduktion von μ möglich, auf die sprunghafte Fehlergröße zu reagieren, so dass es nicht zu Oszillationen des Trainings und folglich der Netzwerkausgabe als Folge einer zu hohen Wahl der Lernrate kommt.

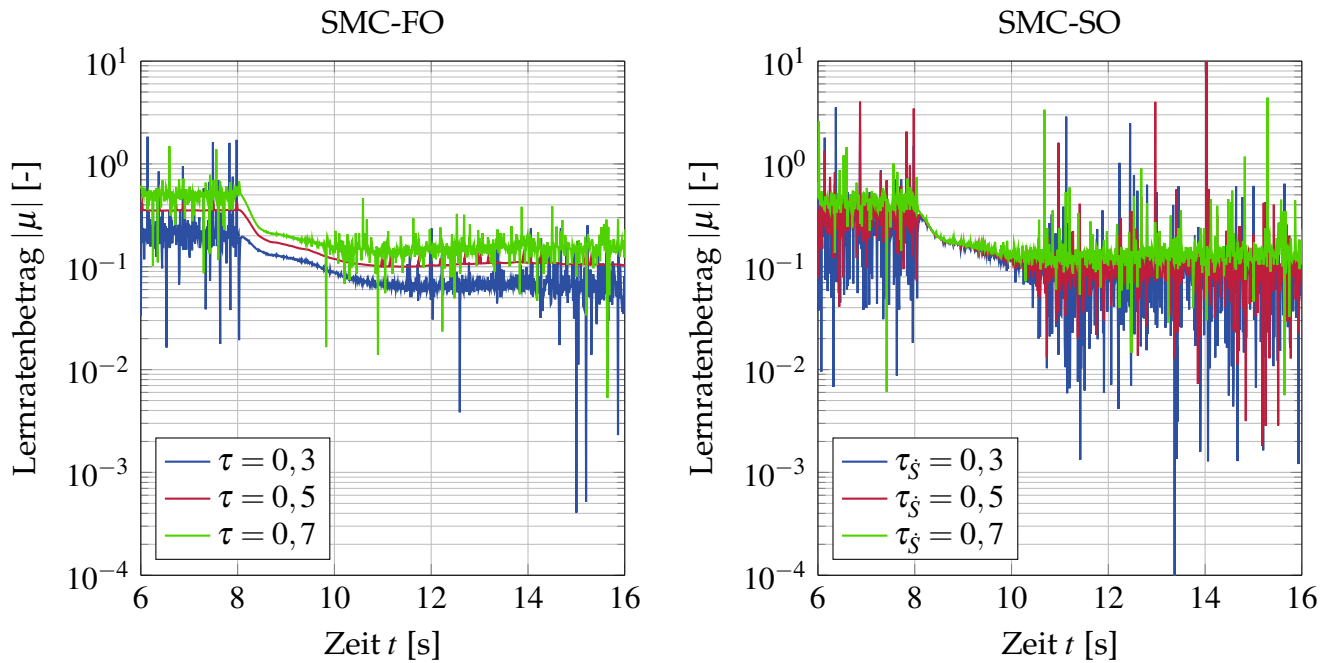


Abbildung 7.6: Lernratenvergleich für den Ausgleich eines sprunghaften Inversionsfehlers bei unterschiedlichen Intervallparametern

Dabei wird in den Abbildungen der Grund für die deutliche Trainingsbeschleunigung bei Änderungen des Intervallparameters durch das Verfahren erster Ordnung, der jedoch nur geringen Verbesserung durch das Verfahren zweiter Ordnung deutlich. Wo für SMC-FO über die gesamte Zeit eine erhöhte Lernratenberechnung für große τ zu erkennen ist, kommt es bei SMC-SO nur vor und nach der Fehleraufschaltung in Sekunde 8 zu sichtbaren Unterschieden in der Lernratenberechnung. Zu Beginn des Fehlerausgleichs wird für alle drei Werte von τ_s eine nahezu identisches μ berechnet, so dass folglich der Fehlerausgleich nur wenig unterschiedlich ausfällt. Dabei ist es beiden Verfahren gemein, dass die Lernratenverläufe nach Aufschaltung des Fehlers und bis zu dessen Ausgleich eine deutlich geringere Dynamik als in der restlichen Zeit der Simulation aufweisen. Dieses Verhalten ist auf die höhere Schaltaktivität in der Nähe der Schaltfunktion bei $S = 0$ zurückzuführen, wie es für einen Fehler e_q von nahezu Null der Fall ist.

Zusammenfassend kann man für den Ausgleich eines sprunghaften Inversionsfehler festgehalten, dass die Ergebnisse aus Kapitel 4 bestätigt werden können. Durch die Berechnung der Lernrate innerhalb stabiler Grenzen kommt es bei der Wahl von $\lambda = 300$ zu einem robusten und schnellen Training durch beide dargestellten Verfahren. Dabei führt die spontane Aufschaltung des Fehlers nicht zu trainingsbedingten Oszillationen, die eine vornherein konservative Auslegung des Netzwerks, wie es bei etablierten Lernverfahren häufig das Vorgehen ist, nötig macht. Im folgenden Abschnitt sollen als letzte Untersuchung des Flugreglers die entwickelten Trainingsverfahren mit dem klassischen Backpropagation-Algorithmus für einen Schadensfall verglichen werden.

7.1.3 Erhebliche Parameterunsicherheiten mit Schwerpunktverschiebung

Für die abschließende Bewertung der Trainingsverfahren wird ein Szenario mit verstärkten Parameterunsicherheiten und einem Schadensfall untersucht. Dabei wird, anders als in Abschnitt 7.1.1, keine einheitliche prozentuale Veränderung der in der Inversion verwendeten Flugzeugparameter vorgenommen, sondern stattdessen eine zufällige Variation genutzt. So werden mittels einer Zufallsverteilung die Modellparameter um bis zu $\pm 50\%$ verändert. Diese zufällige und auf keine direkte physikalische Motivation zurückzuführende Verteilung der Parameteränderungen soll die Charakteristik einer nur sehr rudimentär durchgeführten Systemidentifikation widerspiegeln und bedarf folglich einer robusten Approximation durch die drei neuronalen Netzwerke. Bei dem simulierten Schadensfall handelt es sich um eine nickwinkelabhängige Veränderung des Derivatives der statischen Stabilität $C_{m\alpha}$ durch Multiplikation mit dem Faktor $K_{C_{m\alpha}}$. Motivation für dieses Szenario ist eine gelöste und somit im Flug verschiebbare Masse, die sich in Abhängigkeit der aktuellen Flugzeuglängslage durch den Rumpf bewegt und auf diese Weise die Schwerpunktlage wesentlich beeinflusst. Dabei führt die maximal erreichbare hintere Schwerpunktlage zu einem Vorzeichenwechsel von $C_{m\alpha}$ und somit einer statisch instabilen Flugzeugdynamik um die Querachse. Alle weiteren Einflüsse der Schwerpunktverschiebung auf Hebelarme und Anteile des Trägheitstensors, werden dabei vernachlässigt. So ist es die Aufgabe der neuronalen Netzwerke, neben den erheblichen Parameterunsicherheiten, die zustandsabhängige Änderung der statischen Stabilität auszugleichen und das Flugzeug unter erneutem Einfluss von Wind und Turbulenz zu stabilisieren. In Abbildung 7.7 ist die Ausgabe des Netzwerks der Nickratenregelung für einen ab Sekunde 25 eintretenden Fehlerfall in blau und die Veränderung der statischen Stabilität durch den Faktor $K_{C_{m\alpha}}$ gegenüber der rechten Ordinatenachse in rot aufgetragen. Dabei wurde erneut das Gleitzustandstraining erster Ordnung mit $\lambda = 300$ und $\tau = 0,7$ verwendet.

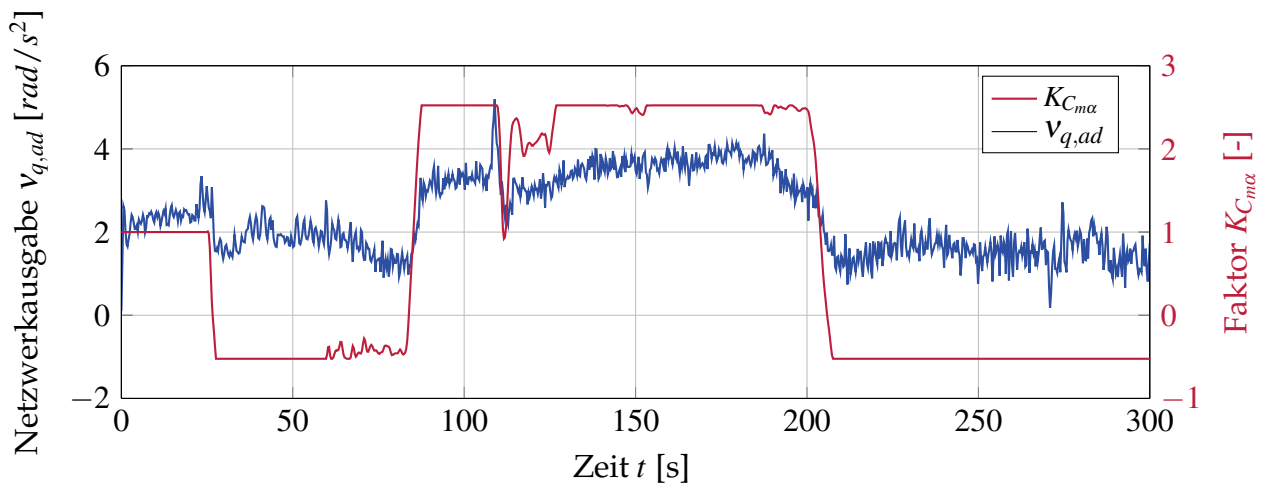


Abbildung 7.7: Netzwerkausgabe und Änderung der statischen Stabilität im Schadensfall

Im Vergleich zur idealen Inversion im letzten Abschnitt ist die von Beginn an höhere Netzausgabe zu erkennen, die notwendig wird, um den bestehenden Inversionsfehler als Folge der erheblichen Modellungenauigkeiten auszugleichen. Dabei ist klar die Auswirkung des bei Sekunde 25 beginnenden Schadensfalls zu sehen. Mit der Veränderung von $K_{C_{m\alpha}}$ und somit der Veränderung der statischen Stabilität als Folge des im Steigflug nach hinten verschobenen Schwerpunkts, ist eine sprunghafte Anpassung der Netzerkennung zu sehen. Dabei verändert sich diese analog der sich einstellenden Schwerpunktlage und gleicht so dauerhaft die Änderung der statischen Stabilität aus. Zur Demonstration der schädlichen Auswirkung der Schwerpunktsverschiebung ist in Abbildung 7.8 der Höhenfehler ΔH des Gleitzustandstrainings erster Ordnung und des konventionellen Backpropagation-Algorithmus (BP) für zwei konstante Lernraten dargestellt.

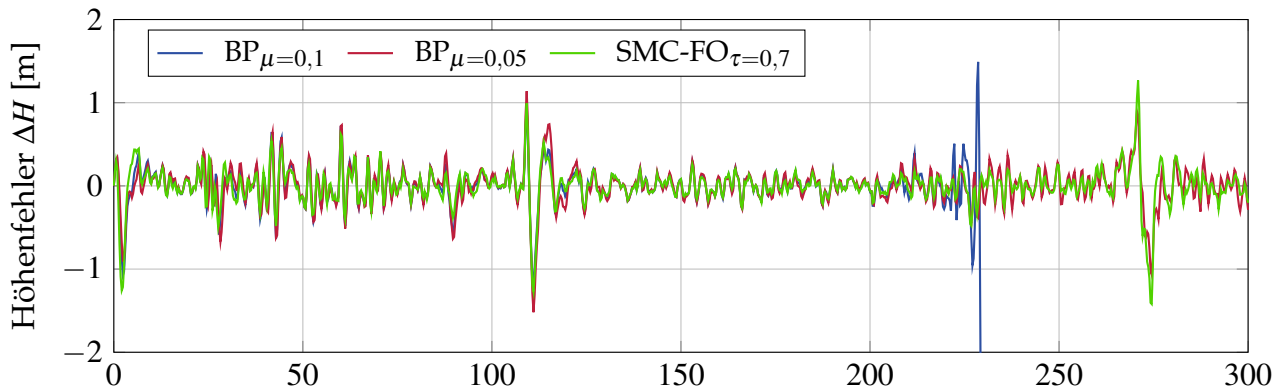


Abbildung 7.8: Vergleich des Höhenfehlers der verschiedenen Lernverfahren im Schadensfall

Wie an der Abbildung zu erkennen ist, kommt es für das Backpropagation-Training mit einer konstanten Lernrate von $\mu = 0,1$ zu Schwingungen in Folge einer oszillierenden Netzerkennung der Rollratenregelung, wie sie in Anhang C.1 zu sehen ist. Dieses destabilisierende Netzwerkverhalten als Reaktion auf eine zu groß gewählte Lernrate führt bei Sekunde 232 letztendlich zum Absturz. Ebenfalls in der Abbildung zu sehen ist, dass für eine entsprechend konservative Lernratenwahl von $\mu = 0,05$ und einer daraus resultierenden leichten Vergrößerung von ΔH im direkten Vergleich zu SMC-FO, der simulierte Schadensfall auch mit dem Backpropagation-Algorithmus ausgeglichen werden kann. Dabei handelt es sich um die höchste empirisch ermittelte Lernrate, bei der die Simulation stabil durchgeführt werden kann. Es ist jedoch nicht möglich, eine Aussage über die Tauglichkeit dieser Lernratenwahl für abweichende Simulationsszenarien zu treffen.

Für die folgende Untersuchung der Trainingsalgorithmen wird das beschriebene Schadenszenario einer dynamisch veränderlichen Schwerpunktlage mit dem oben vorgestellten Fall eines induzierten Inversionsfehlers kombiniert. Auf diese Weise soll durch den auf der einen Seite sprunghaften Anstieg des Inversionsfehlers und dem auf der anderen Seite dauer-

haften Bedarf nach kontinuierlicher Anpassung an die dynamische Schwerpunktlage die Robustheit der vorgestellten Trainingsverfahren untersucht werden. In Abbildung 7.9 sind zu diesem Zweck erneut die Höhenfehlerverläufe ΔH für das Gleitzustandstraining erster Ordnung mit $\lambda = 300$ und den Backpropagation-Algorithmus mit der zuvor bestimmten Lernrate von $\mu = 0,05$ dargestellt.

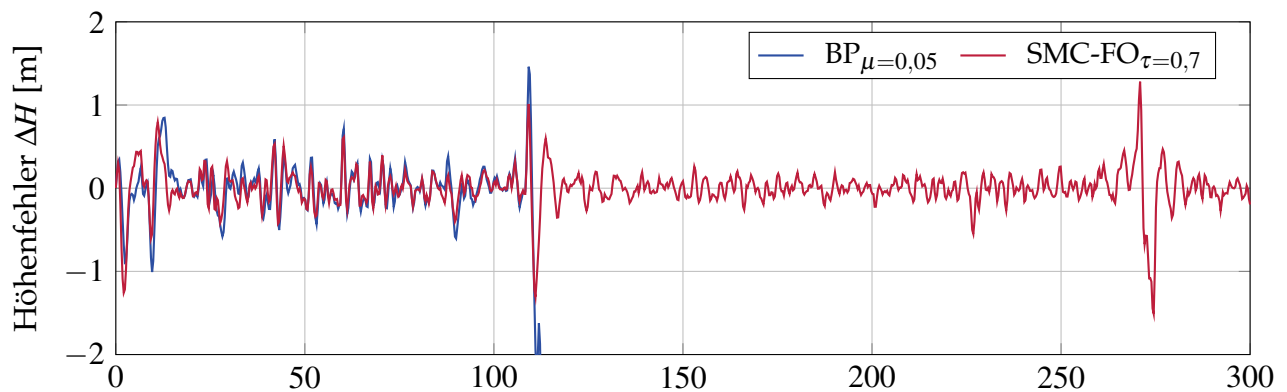


Abbildung 7.9: Vergleich des Höhenfehlers der beiden Lernverfahren im Schadensfall mit zusätzlichem sprunghaftem Inversionsfehler

Man erkennt, dass die zuvor getroffene konservative Lernratenwahl für den konventionellen Backpropagation-Algorithmus nicht mehr ausreicht, um die Kombination verschiedener Fehler und Schäden stabil über die gesamte Flugzeit auszugleichen. Darüber hinaus konnte keine konstante Lernrate bestimmt werden, die hoch genug ist, um den induzierten Inversionsfehler dauerhaft zu kompensieren, dabei jedoch gleichzeitig gering genug ist, um schädliche Oszillationen zu vermeiden. Im Vergleich dazu führt das Gleitzustandstraining erster Ordnung zu einem stabilen und robusten Trainingsergebnis und somit zum Ausgleich der aufgeschalteten Fehler- und Schadensfälle. An den Verläufen der Frobeniusnorm in Anhang C.2 ist gut zu erkennen, dass es dem Gleitzustandstraining durch die dynamische Lernratenbestimmung schneller möglich ist, auf die aufgeschalteten Störungen zu reagieren und die Netzwerkgewichte in kürzerer Zeit so zu verändern, dass eine Ausgabe erzeugt wird, die zu einem stabilen Fehlerausgleich führt.

Zum abschließenden Nachweis der Leistungsfähigkeit der entwickelten Gleitzustandsverfahren mit Verwendung in einer Flugregelungsanwendung sind in Tabelle 7.1 die mittleren quadratischen Fehler der Nick- und Rollratenregelung für verschiedene Trainingsparameterkombinationen aufgeführt. Dabei soll durch die Spanne der gewählten Parametersätze die Robustheit der auf Grundlage von Erreichbarkeits- und Existenzbedingungen beruhenden Lernratenbestimmung demonstriert werden. Zu diesem Zweck sind sowohl Ergebnisse für die Kombination hoher als auch niedriger Werte von Intervall- und Schaltfunktionsparameter dargestellt.

Verfahren	λ	τ	$\tau_{\dot{s}}$	$MSE_q [(\circ/s)^2]$	$MSE_p [(\circ/s)^2]$
SMC-FO	100	0,3	-	6,1761	38,1843
SMC-SO	100	0,3	0,3	6,5610	40,8844
SMC-FO	300	0,7	-	3,4992	23,3494
SMC-SO	300	0,7	0,7	3,4564	23,5882
SMC-FO	600	0,7	-	3,3509	21,1101
SMC-SO	600	0,7	0,7	3,3117	19,8623

Tabelle 7.1: Fehlerauswertung im Schadensfall mit Inversionsfehler, Parameterunsicherheiten und atmosphärischen Störungen

Die Kombination kleiner Werte für sowohl λ als auch τ und $\tau_{\dot{s}}$ hat wie zu erwarten eine Reduktion der Regelgüte zur Folge, ohne dabei aber die Konvergenzgeschwindigkeit so weit abzusenken, dass der Ausgleich der simulierten Fehlerfälle misslingt. Auch für die Simulationen mit einer Kombination von hohen Intervall- und Schaltfunktionsparametern werden die drei verwendeten neuronalen Netzwerke stabil und schwingungsfrei trainiert. So kann festgehalten werden, dass die Wahl der einstellbaren Netzwerkparameter innerhalb der in Kapitel 4 aufgestellten Grenzen zwar eine Auswirkung auf die generelle Trainingsgüte, jedoch nicht auf die Stabilität hat. Darüber hinaus legen die Ergebnisse aus Tabelle 7.1 erneut die Wahl des Schaltfunktionsparameters $\lambda = 300$ nahe, da bei der Erhöhung von $\lambda = 100$ auf diesen Wert noch eine deutliche Verbesserung der mittleren quadratischen Fehler realisiert werden kann, eine anschließende Verdopplung auf $\lambda = 600$ gerade in der Nickratenregelung jedoch kaum noch eine Auswirkung zeigt.

Zusammenfassend kann für die Untersuchung des Flugreglers die Motivation für die Verwendung eines Gleitzustandstrainings aus Kapitel 4 bestätigt werden. Durch die empirische Wahl der Lernrate für den Backpropagation-Algorithmus kann zwar oftmals ein erfolgreiches und auch leistungsstarkes Training durchgeführt werden, jedoch besteht die Unsicherheit, ob die ermittelte und für das aktuelle Szenario performante Trainingsrate in einem veränderten Flugzustand nicht zu Instabilitäten führt. Dahingegen kann gezeigt werden, dass die dynamische Lernratenberechnung der entwickelten Verfahren unabhängig von der Wahl der Parameter zu stabilen Trainingsergebnissen führt. Dabei erfordert eine Veränderung der Simulationsrandbedingungen keine Anpassung der Netzwerkparameter, da durch die den Trainingsverfahren zugrunde liegenden Erreichbarkeits- und Existenzbedingungen zu jedem Zeitschritt stabile Grenzen für das Netzwerktraining bestimmt werden können.

7.2 Fahrsimulation

In diesem Abschnitt soll der zweite vorgestellte Inversionsregler für die Querdynamikregelung eines Fahrzeugs untersucht werden. Dabei erlaubt der als Bahnvorgabe genutzte Rundkurs aus Abbildung 7.10 den Vergleich mehrerer Runden und die Untersuchung des Langzeitverhaltens des implementierten neuronalen Netzwerks. Das kommandierte Geschwindigkeitsprofil für die Simulation der automatischen Bahnfolge ist in Anhang C.3 über den sogenannten Referenzpunkten s_{ref} aufgetragen. Diese werden für die Referenzierung einer Position auf dem Rundkurs verwendet und geben die auf den Startpunkt bezogene zurückgelegte Strecke in Metern an. Die initiale Geschwindigkeit des simulierten Fahrzeugs zum Zeitpunkt $t_0 = 0$ ist dabei ebenfalls null, so dass erst in der zweiten Runde das gezeigte Geschwindigkeitsprofil vollständig abgefahren wird. Für die Untersuchung des Fehlerausgleichs am Beispiel des Fahrzeugreglers wird in Abschnitt 7.2.2 das Platzen des vorderen linken Reifens bei Sekunde 25 simuliert. Neben dem Startpunkt der Simulation und der Fahrtrichtung ist in Abbildung 7.10 zusätzlich der Punkt des aufgeschalteten Reifenplatzens markiert.

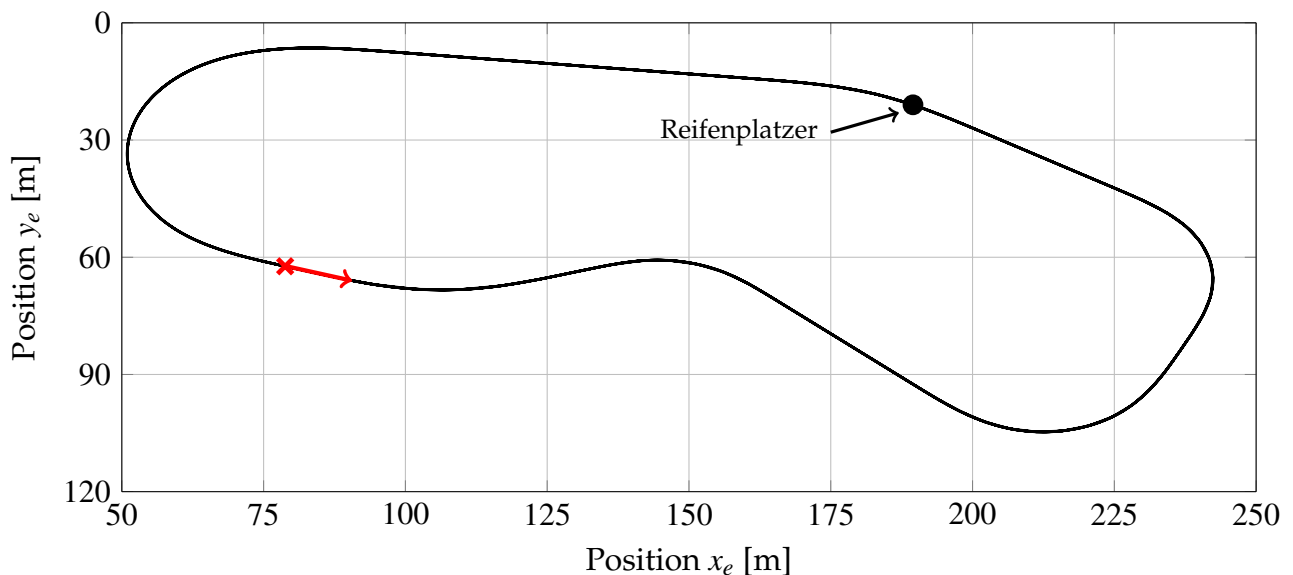


Abbildung 7.10: Rundkurs für die Untersuchung des Fahrzeugreglers

7.2.1 Langzeituntersuchung mit leichten Parameterunsicherheiten

Für die Untersuchung des rundenweisen Einflusses des neuronalen Netzwerks auf das Regelungsergebnis im schadenfreien Nominalfall und der Auswirkung eines kontinuierlichen Trainings wird eine 60 minütige Simulation mit insgesamt 92 Runden durchgeführt. Dabei wird erneut ein künstlicher Inversionsfehler durch eine zufällige Erhöhung bzw. Reduktion der für die Inversion verwendeten Modellparameter um 25% umgesetzt. So kann neben der Kompensation der starken Reifenmodellvereinfachung in der Steuergrößenzuweisung (vgl. Abschnitt 6.2.2) der Ausgleich eines konstanten Inversionsfehlers über mehrere Runden untersucht und bewertet werden. Auf Grundlage der vorangegangenen Simulationen

des Flugreglers wird für das in den folgenden Bewertungen verwendete Gleitzustandstraining erster Ordnung $\lambda = 300$ und $\tau = 0,7$ gewählt. Die Ergebnisse von vier beispielhaft ausgewählten Runden für den Gierratenfehler $e_{\dot{\psi}}$ und somit der direkt durch das neuronale Netzwerk beeinflussten Rotationsdynamik sind in Abbildung 7.11 über s_{ref} aufgetragen.

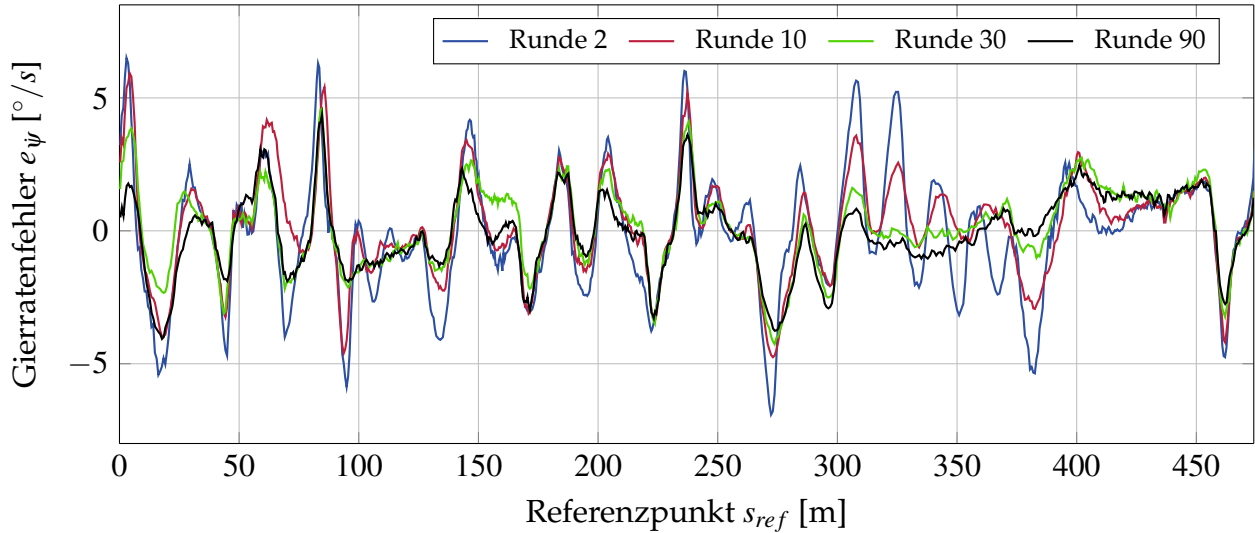


Abbildung 7.11: Fehlervergleich der Gierratenregelung für mehrere Runden mit SMC-FO

Wie es für den Einsatz eines *online*-trainierten neuronalen Netzwerks zu vermuten ist, kann das Regelungsergebnis der Rotationsdynamik über die steigende Anzahl an gefahrenen Runden verbessert werden. So ist nach der zehnten Runde eine klare Reduktion des Fehlers gegenüber dem dargestellten Verlauf der zweiten Runde zu sehen. Dies gilt insbesondere für den Streckenabschnitt zwischen Referenzpunkt 300 und 400. Das fortlaufende Netzwerktraining führt im Anschluss bei Runde 30 zu einer vollständigen Vermeidung der zuvor zu sehenden Überschwinger in diesem Streckenbereich und einer generell gestiegenen Regelgüte. Weitere 60 Runden später kommt es zu nur noch leichten Verbesserungen der Gierratenregelung, so dass das Netzwerktraining in der Langzeituntersuchung ein gegen ein Optimum konvergierendes Verhalten zeigt. Der Verlauf des mittleren quadratischen Fehlers MSE_r für die einzelnen Runden ist in Anhang C.4 abgebildet. Die in den vier beispielhaft untersuchten Runden erreichte seitliche Bahnabweichung d , als generelles Gütekriterium für die Querdynamikregelung, ist in Abbildung 7.12 dargestellt. Man kann erkennen, dass die bereits in der zweiten Runde sehr hohe Qualität der simulierten Bahnführung mit zunehmender Rundenanzahl noch weiter ansteigt. So kann, wie schon in Abbildung 7.11 zu sehen, das leicht schwingende Fahrzeugverhalten ab Referenzpunkt 300 zunächst reduziert und im Anschluss vollständig abgebaut werden.

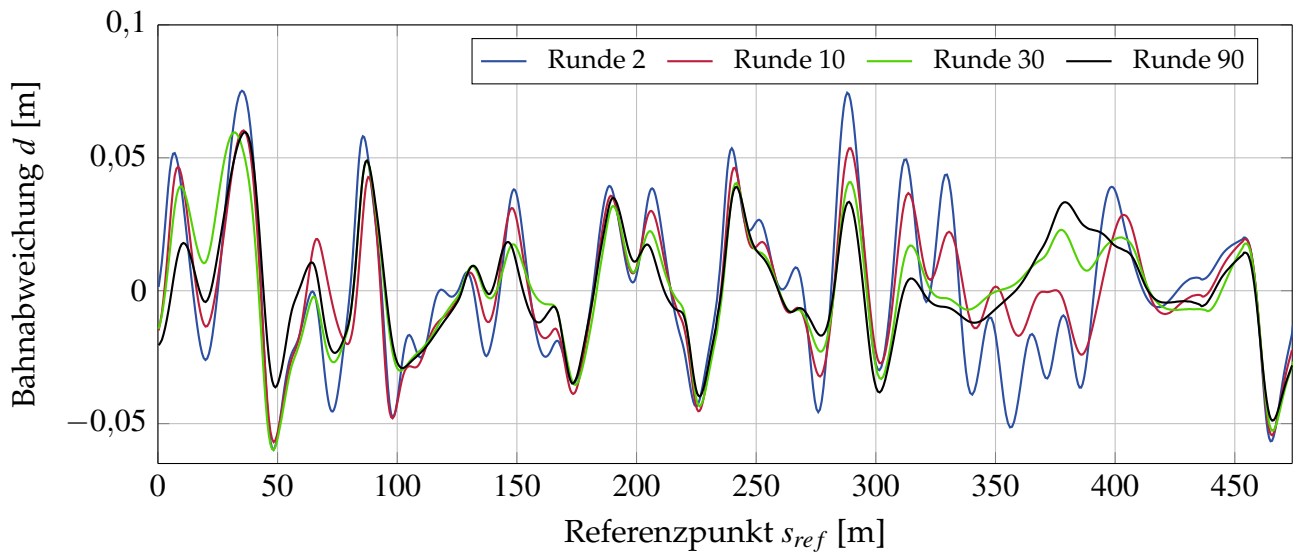


Abbildung 7.12: Vergleich der Bahnabweichung für mehrere Runde mit SMC-FO

Interessant zu beobachten ist, dass infolge des Netzwerktrainings die Bahnabweichung ab Referenzpunkt 350 ein anderes Vorzeichen annimmt, so dass durch die Approximation der simulierten Inversionsfehler die Kurve augenscheinlich auf einer veränderten Linie durchfahren wird. So werden für die Runden 30 und 90 zunächst positive Abweichungen aufgebaut, welche letztendlich zu einem beschleunigten Fehlerabbau beim Kurveneintritt am Referenzpunkt 400 führen, was einer veränderten Einlenkstrategie für diese Kurve entspricht. Dieses Verhalten unterstreicht anschaulich den Wissensaufbau im neuronalen Netzwerk.

Die in Abbildung 7.11 und 7.12 gezeigte zunächst schnelle und mit steigender Rundenanzahl nur noch in kleinen Schritten stattfindende Anpassung durch das neuronale Netzwerk kann auch der in Abbildung 7.13 dargestellten Frobeniusnorm entnommen werden.

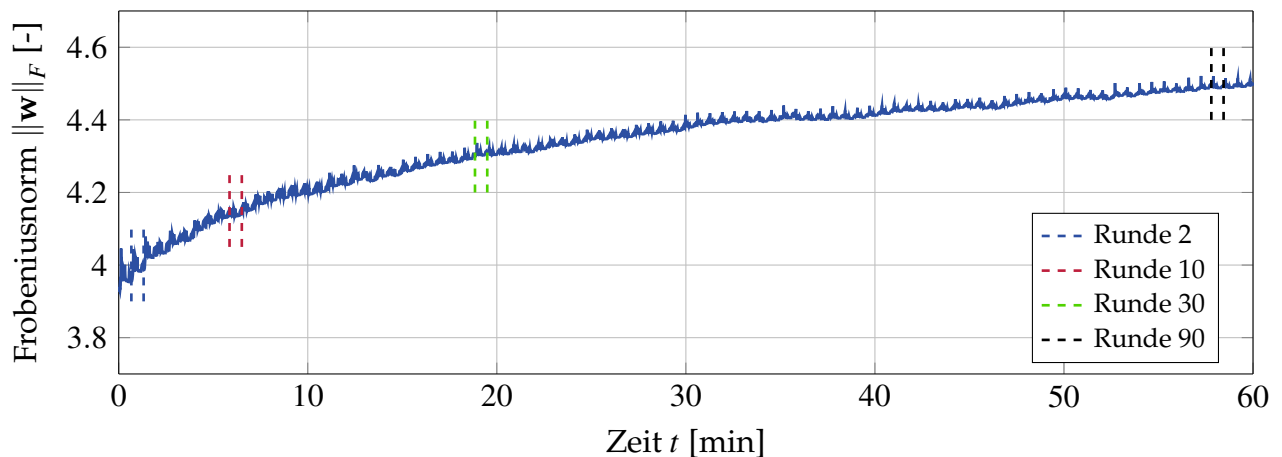


Abbildung 7.13: Frobeniusnorm für das Langzeittraining mit SMC-FO

Zur besserer Orientierung sind der Start und das Ende der vier oben beispielhaft betrachteten Runden für den über der Zeit dargestellten Normverlauf markiert. So kann einer jeweiligen Runde direkt das entsprechende Normniveau zugeordnet werden. Es ist klar ein zu Beginn der Simulation beschleunigtes Wachstum der Netzwerknorm zu erkennen, bevor der Anstieg mit fortschreitender Simulationsdauer abflacht. So kommt es in den ersten 30 Runden zu einem absoluten Wachstum der Norm von ungefähr 0,4 und in einer doppelt so langen Zeit von 60 weiteren Runden nur noch zu einem Wachstum von ungefähr 0,2. Dieses Verhalten kann dabei zusätzlich durch die Parametrierung des robustifizierenden Terms beeinflusst werden. Durch die Tendenz des neuronalen Netzwerks bei sprunghaften Fehlern mit einer ebenfalls sprunghaften Normerhöhung zu reagieren, kann ein relativ zu den übrigen Pseudosteuergrößenanteilen großer robustifizierender Term dazu verwendet werden, die Auswirkung solcher Fehler auf das neuronale Netzwerk zu dämpfen und so einen dauerhaften Anwachsen der Norm zu verhindern. Dies geschieht, indem die sprunghafte Störung initial durch ein großes v_r ausgeglichen wird, so dass es zu einer weniger ausgeprägten Reaktion der Netzwerknorm infolge des gedämpften Fehlereinflusses kommt. Die aus dem hier dargestellten Normverlauf resultierenden Netzwerkausgaben der vier betrachteten Runden sind in Anhang C.5 dargestellt. Dabei ist erneut die Reduktion der leichten Schwingungen ab Referenzpunkt 300 zu erkennen.

Zum Vergleich der Leistungsfähigkeit des Gleitzustands- und des Backpropagation-Trainings sind in Tabelle 7.2 die mittleren quadratischen Fehler der Gierratenregelung sowie der seitlichen Bahnablage dargestellt. Dabei ist es für den konventionellen Trainingsansatz erneut notwendig, eine empirische Lernratenbestimmung durchzuführen, wohingegen für die Gleitzustandsverfahren die Intervall- und Schaltfunktionsparameter aus den Untersuchungen des Flugreglers übernommen werden können.

Verfahren	τ	τ_s	μ	$MSE_r [(\circ/s)^2]$	$MSE_d [m^2]$
BP	-	-	0,1	1,9646	0,000549
SMC-FO	0,7	-	-	2,6998	0,000473
SMC-SO	0,7	0,7	-	2,7906	0,000471

Tabelle 7.2: Fehlerauswertung für 60 minütigen fehlerfreien Nominalfall mit leichten Parameterunsicherheiten

Der Tabelle kann entnommen werden, dass alle drei Verfahren zu einem leistungsstarken Regelungsergebnis für die automatische Bahnfolge über 60 Minuten führen. Dabei kann jedoch für das Backpropagation-Verfahren nicht mit Sicherheit gesagt werden, ob die ermittelte Lernrate auch im Schadensfall zu einem stabilen Netzwerkverhalten führt. Im folgenden Abschnitt soll aus diesem Grund die Netzwerkreaktion nach dem simulierten Platzen eines Reifens untersucht werden.

7.2.2 Netzwerkverhalten bei Reifenplatzer

Die Simulation des für die Untersuchung des Fehlerausgleichs verwendeten Reifenplatzers wird in dieser Arbeit an die Beschreibung aus [142] angelehnt. So kommt es beim Platzen des vorderen linken Reifens innerhalb einer Zeitspanne von $\Delta T = 0,1$ Sekunden zu dem Aufbau einer der Fahrtrichtung entgegenwirkenden Kraft $F_{x,Widerstand}$ von 2000N, sowie der prozentualen Verringerung der erzeugbaren seitlichen Reifenkraft F_y und des Rückstellmoments M_z um 75%. Zur Veranschaulichung ist dieses Prinzip der sich einstellenden Reifenkraft- und Momentenänderung schematisch in Abbildung 7.14 dargestellt.

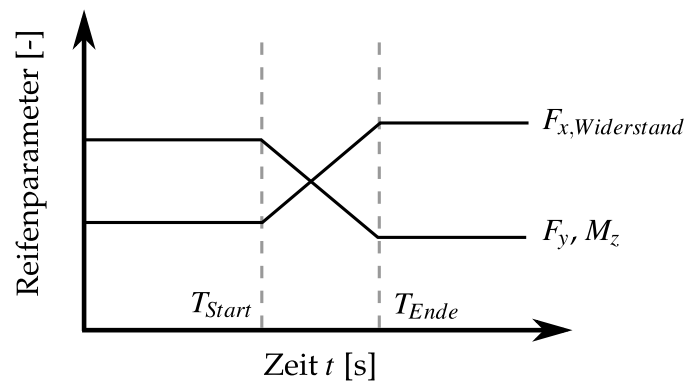


Abbildung 7.14: Änderung der Reifenkräfte und -momente in Folge eines Reifenplatzers nach [142]

Zusätzlich wird in der Simulation zum Zeitpunkt des Reifenplatzers der Radsturz am betroffenen vorderen linken Rad um -12° und die Spur um 8° verändert, sowie ein Lenkwinkelsprung von 100° aufgegeben. Dabei stellen diese angenommenen Werte eine zusätzliche Steigerung des degradierenden Einflusses des geplatzten Reifens auf die Fahrzeugdynamik dar. So handelt es sich bei dem dargestellten Szenario um einen kritischen Schadensfall, der einer schnellen Kompensation durch das neuronale Netzwerk bedarf, um eine stabile Fortführung der automatischen Fahrt sicherstellen zu können. Um der stark verminderten Fahrzeugdynamik Rechnung zu tragen, wird eine Sekunde nach Auftreten des Reifenschadens das vorgegebene Geschwindigkeitsprofil um 40% reduziert.

In Abbildung 7.15 sind die seitliche Bahnabweichung der drei untersuchten Lernverfahren abgebildet. Dabei musste die konstante Lernrate des konventionellen Trainingsalgorithmus gegenüber dem Nominalfall angepasst werden, um den kritischen Reifenschaden bei Sekunde 25 ausgleichen zu können. Der Abbildung ist zu entnehmen, dass alle drei untersuchten Lernverfahren dazu in der Lage sind, den Reifenplatzer auszugleichen und die Fahrt mit reduzierter Fahrtgeschwindigkeit fortzusetzen. Dabei ist jedoch zu sehen, dass die beiden simulierten Gleitzustandsverfahren deutlich weniger Schwingungen um die Nulllage nach Aufschalten des Fehlers aufweisen. So kommt es zu einem reduzierten Fehlermaximum und einem ruhigeren Ausgleich der aufgebauten Bahnabweichung im Vergleich zum Backpropagation-Algorithmus, ohne dass es notwendig wird Änderungen an den Netzwerkparametern im Vergleich zum Nominalfall vorzunehmen.

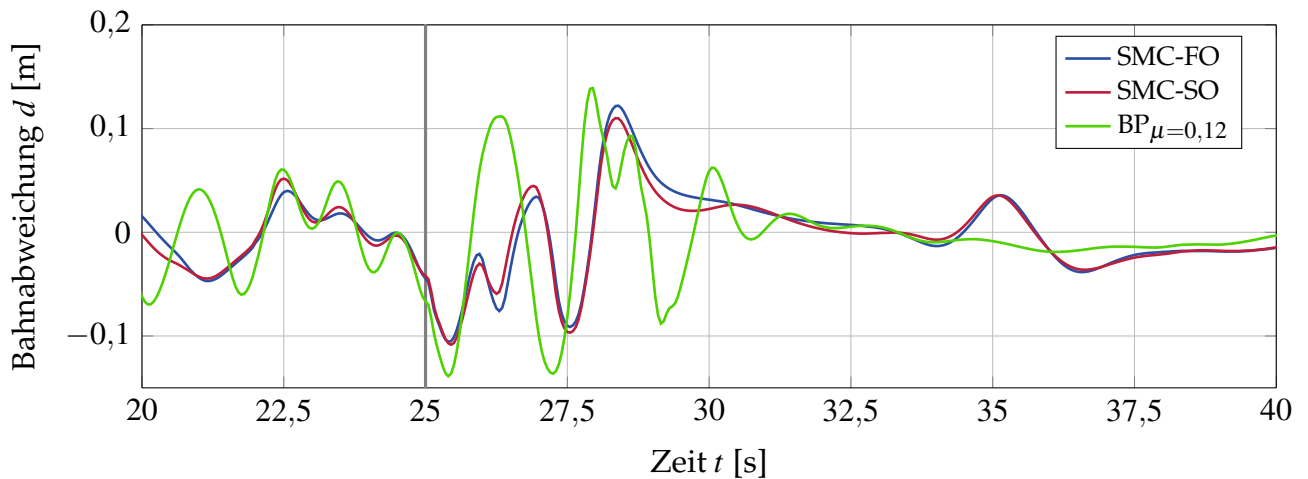


Abbildung 7.15: Vergleich der Bahnabweichung für unterschiedliche Lernverfahren mit Reifenplatzer bei Sekunde 25

Zusammenfassend belegen die Simulationen des Fahrzeugreglers für den Nominal- und Schadensfall die Ergebnisse der Flugregelungsuntersuchung. Die in Kapitel 4 bestimmten und in der Simulation des Flugreglers bereits untersuchten Netzwerkparameter für die Gleitzustandsverfahren erster und zweiter Ordnung führen auch für das Szenario einer einständigen automatischen Fahrt zu einer leistungsstarken und stabilen Regelung. Dabei ist es nicht notwendig, die Intervall- und Schaltfunktionsparameter der entwickelten Lernverfahren für den Schadensfall zu ändern, was die Robustheit der vorgestellten dynamischen Lernratenbestimmung und die Stabilität des daraus resultierenden Netzwerktrainings noch einmal unterstreicht.

7.3 Fahrversuch

Nach den oben gezeigten Untersuchungen des Fahrzeugreglers in der Simulation, sollen in diesem Abschnitt Ergebnisse praktischer Fahrversuche dargestellt werden. Dabei konnten aufgrund zeitlicher Limitierungen bei der Beanspruchung des Versuchsträgers nur Untersuchungen zum Gleitzustandstraining erster Ordnung mit deaktiviertem PCH durchgeführt werden. Eine weitere Unterscheidung zu den simulativen Untersuchungen stellt die Wahl konstanter Eingangsgrößen von Eins für das neuronale Netzwerk dar. Motiviert ist dieses Vorgehen durch ein stetiges Wachstum der Netzwerkgewichte für dynamische Eingangsgrößen, was unter anderem auf eine an den Versuchstagen fehlenden Eingangsgrößenskalierung und einer vernachlässigbar geringen Stützung durch den robustifizierenden Term zurückzuführen ist. Durch die Wahl konstanter Eingangsgrößen und der daraus folgenden Entkopplung der Vorwärtspropagation von der Systemdynamik kann dieses schädliche Verhalten verhindert werden. Ein Vorteil dieser Umsetzung ist die schon in Kapitel 4 beschriebene Reduktion der netzwerkbeeinflussenden Größen und der daraus resultierenden Fokussierung auf die Adaptionsgeschwindigkeit des verwendeten Lernverfahrens. Dem Netzwerk ist es durch die Vorgabe konstanter Eingangsgrößen jedoch nicht möglich,

Wissen aufzubauen, so dass die maßgeblichen Bewertungskriterien der gezeigten Fahrversuche die Geschwindigkeit des Lernverfahrens sowie die Robustheit und Stabilität der Lernratenbestimmung sind. Eine generelle Aussage zum dauerhaften Anlernen eines vorliegenden Inversionsfehlers, so dass das neuronale Netzwerk einen Zusammenhang zwischen der Fahrzeugdynamik und dem jeweiligen Trainingsfehler herstellt, kann jedoch nicht getroffen werden.

7.3.1 Fehlerausgleich auf einfachem Rundkurs

Für die erste Bewertung des adaptiven Inversionsreglers im Fahrversuch wird ein einfacher Rundkurs, wie er in Abbildung 7.16 zu sehen ist, gefahren. Dabei wird als künstlich erzeugter Schadensfall ein konstanter Lenkradwinkelsprung von 120° an der Stelle $s_{ref} = 50$ aufgeschaltet und bei $s_{ref} = 250$ wieder vom Lenksignal abgezogen. Die erheblichen Störungen der automatischen Bahnfolge bei sowohl dem Start als auch dem Ende der Fehleraufschaltung machen eine schnelle Kompensation durch das Regelungssystem notwendig. Wie eingangs beschrieben, wurde für die Fahrversuche ausschließlich das Gleitzustandstraining erster Ordnung verwendet. Dabei wurde für die Parameter des neuronalen Netzwerks, aufgrund der oben beschriebenen Umsetzungsunterschiede zur Simulation und einer generell konservativen Auslegung bei der Inbetriebnahme, $\lambda = 50$ und $\tau = 0,5$ gewählt. Ebenfalls wurden die Regelungsparameter für die Verwendung im Fahrversuch angepasst, was den Unterschieden zwischen der simulierten und realen Regelungsstrecke Rechnung trägt.

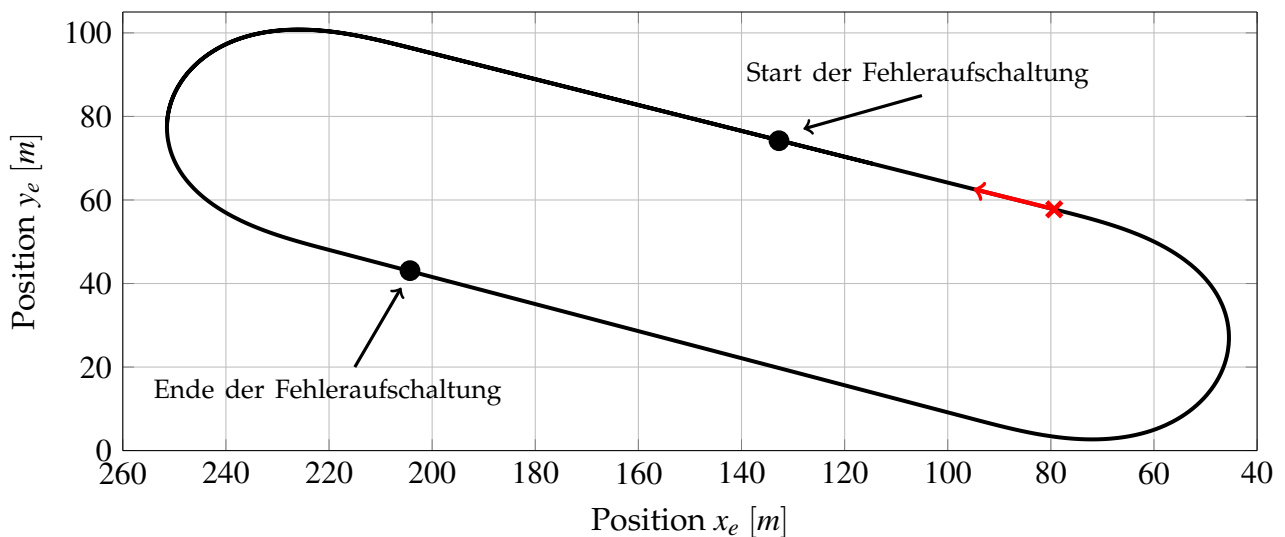


Abbildung 7.16: Einfacher Rundkurs für die Untersuchung des Fahrzeugreglers im Fahrversuch

Für die im Anschluss gezeigten Ergebnisse wurden sechs aufeinander folgende Runden des dargestellten Rundkurses gefahren. Dabei wurde die Geschwindigkeitsvorgabe nach jeweils zwei Runden gesteigert. Auf diese Weise kann die Auswirkung höherer Geschwindigkeiten sowohl auf die generelle Bahnführungsgenauigkeit als auch auf den Fehlerausgleich untersucht werden. Dabei wird die Geschwindigkeitsregelung für die vollautomatische Fahrt,

wie bereits erwähnt, durch einen nicht in dieser Arbeit entwickelten PID-Regler umgesetzt. Zur Skalierung des vorgegebenen Geschwindigkeitsprofils wird der Faktor V_{Skal} eingeführt. Mit dessen Hilfe lässt sich die mit dem verwendeten Fahrzeug maximal umsetzbare Geschwindigkeitsvorgabe prozentual verringern, wie es in Abbildung 7.17 für die drei verwendeten Werte von V_{Skal} mit Markierung der Fehlerauf- und -abschaltung dargestellt ist.

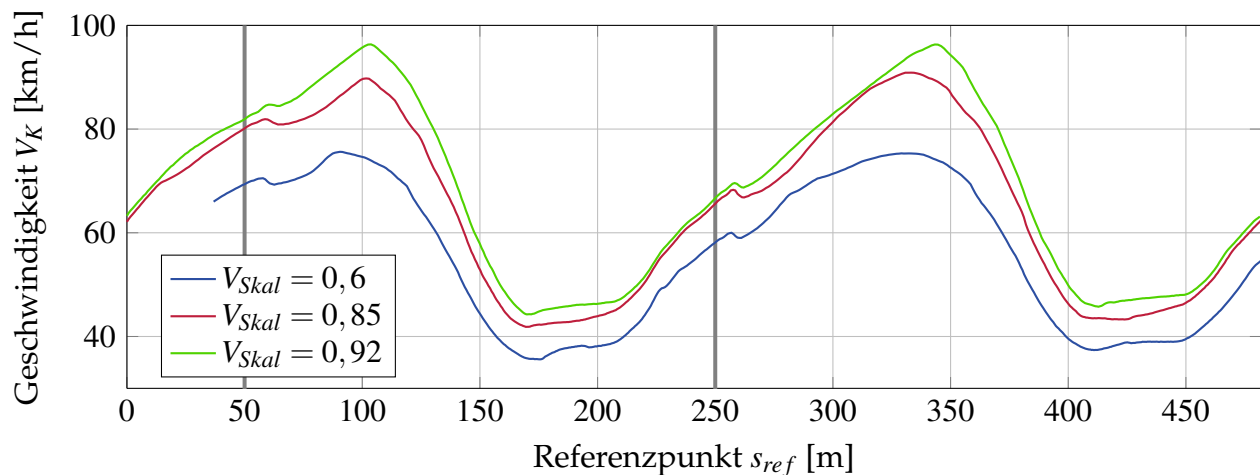


Abbildung 7.17: Skalierte Geschwindigkeitsvorgabe für die automatische Fahrt mit Fehleraufschaltung

Die für die drei dargestellten Geschwindigkeitsverläufe gemessenen Querbeschleunigungen sind als Referenz der dynamischen Belastung in Anhang C.6 abgebildet. Dabei sind klar die Auswirkungen der Fehlerauf- und -abschaltung in Form sprunghaft aufgebauter Querbeschleunigungen zu erkennen. Für die Fahrt mit $V_{Skal} = 0,92$ kommt es in den Kurven zu Querbeschleunigungen von bis zu 8 m/s^2 , so dass die umgesetzte Bahnvorgabe zu Zuständen im fahrdynamischen Grenzbereich führt.

Das Ergebnis der automatischen Fahrt mit dem Inversionsregler und einem gleitzustands-trainierten neuronalen Netz sind in Form der seitlichen Bahnabweichung d in Abbildung 7.18 dargestellt. Dabei ist zur besseren Übersicht jeweils nur eine Runde für die Fahrt mit einem vorgegebenen Wert von V_{Skal} dargestellt. Es ist klar zu sehen, dass für alle drei verwendeten Geschwindigkeitsvorgaben der Lenkradwinkelsprung stabil ausgeglichen wird und die automatische Fahrt im Anschluss sicher fortgeführt werden kann. Dabei kommt es, wie zu erwarten, für $V_{Skal} = 0,92$ zu den höchsten seitlichen Bahnabweichungen von bis zum 0,8 Metern. Nach Abschaltung des Fehlers wird das Fahrzeug für alle drei Geschwindigkeiten mit hoher Genauigkeit zurück zur Bahn geführt, so dass es zu gemessenen Bahnabweichungen im einstelligen Zentimeterbereich kommt. Damit kann festgehalten werden, dass für die Verwendung von konstanten Eingangsgrößen die zweimalige sprunghafte Kompensation eines Lenkradwinkelsprungs pro Runde keinen schädlichen Einfluss auf die weitere Bahnführung mit Stützung durch das neuronale Netzwerk hat.

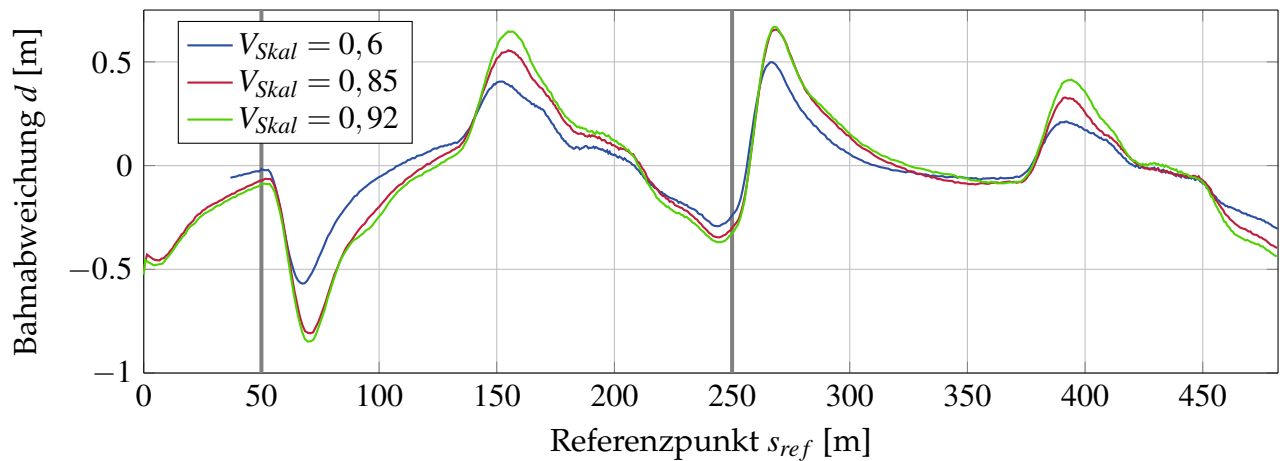


Abbildung 7.18: Vergleich der Bahnabweichungen für die unterschiedlichen Geschwindigkeitsvorgaben mit Fehleraufschaltung

Die für die Fahrt mit $V_{Skal} = 0,92$ erzeugte Netzwerkausgabe v_{ad} , der robustifizierenden Term v_r sowie die Ausgabe des Fehlerreglers v_e sind in Abbildung 7.19 dargestellt. Dabei ist klar der im Vergleich zur Simulation deutlich abgeschwächte Verlauf für v_r zu sehen, der auf diese Art nahezu keinen Einfluss auf den Fahrversuch hat und ein Grund für die Wahl konstanter Eingabegrößen des Netzwerks ist.

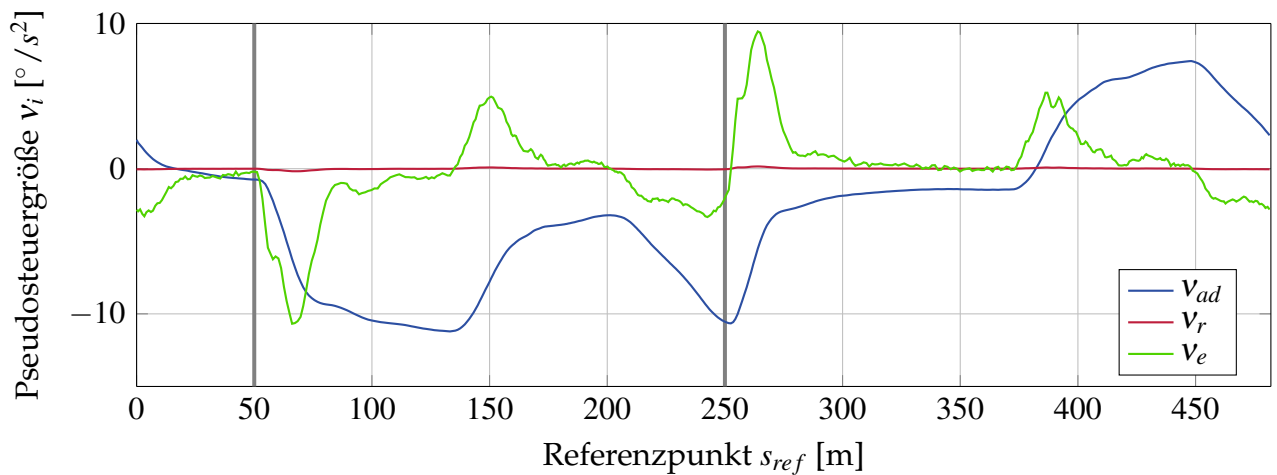


Abbildung 7.19: Vergleich der Pseudosteuergrößenanteile für die automatische Fahrt mit Fehleraufschaltung bei $V_{Skal} = 0,92$

Anhand der Netzwerkausgabe v_{ad} ist deutlich die Kompensation des aufgeschalteten Fehlers und die Unterstützung in der Kurvenfahrt zu erkennen. Im Anschluss an die Fehleraufschaltung bei Referenzpunkt 50 kommt es zunächst zu einer sprunghaften Reaktion des Linearreglers, bevor das neuronale Netzwerk den Lenkradsprung durch eine angepasste Ausgabe kompensiert. Daran anschließend kommt es für die Kurvenfahrt zu einer Reduktion der Netzausgabe, bis diese nach Verlassen der Kurve wieder auf den Wert der Fehlerkompensation ansteigt. Nach Abschaltung des Fehler bei Referenzpunkt 250

nimmt die Netzausgabe über längere Zeit wieder einen gleichbleibenden Wert an, wobei dieser ungleich Null ist und somit für den dauerhaft Ausgleich eines Inversionsfehlers steht. Damit kann das Ergebnis aus der Simulation bestätigt werden, dass durch das neuronale Netzwerk der Linearregler soweit entlastet werden kann, dass dieser nur bei den aufgetragenen Lenkradsprüngen und dem Ein- und Ausleiten von Kurven einen größeren Anteil an der Pseudosteuergröße v stellt. Im übrigen Verlauf kommt es somit zu Werten von v_e um die Nulllage herum.

Zum Vergleich des adaptiv gestützten Inversionsreglers mit einem rein robusten Regelungsansatz konnten Fahrten mit einem leistungsstarken und für die automatische Bahnfolge im dynamischen Grenzbereich umfassend getesteten Referenzregler durchgeführt werden. Dieser führt für das fehlerfreie Szenario zu Regelungsergebnissen von sehr hoher Güte. Somit ist sichergestellt, dass die beiden Regelungsansätze über eine vergleichbare Leistungsfähigkeit im Nominalfall verfügen, was die Grundlage für die Auswertungen im Schadensfalls darstellt. Da es sich bei dem Referenzregler um einen rein robusten, jedoch nicht adaptiven Ansatz handelt, wurde der aufgeschaltete Lenkradsprung auf einen Wert von 80° reduziert. Zusätzlich wurde für die Geschwindigkeitsskalierung ein ebenfalls reduzierter Wert von $V_{Skal} = 0,5$ verwendet. Die Bahnablage des Referenzreglers (Ref) im Vergleich zu der bereits dargestellten Fahrt des Inversionsreglers (Inv) mit $V_{Skal} = 0,6$ und Lenkradsprung von 120° ist in Abbildung 7.20 dargestellt.

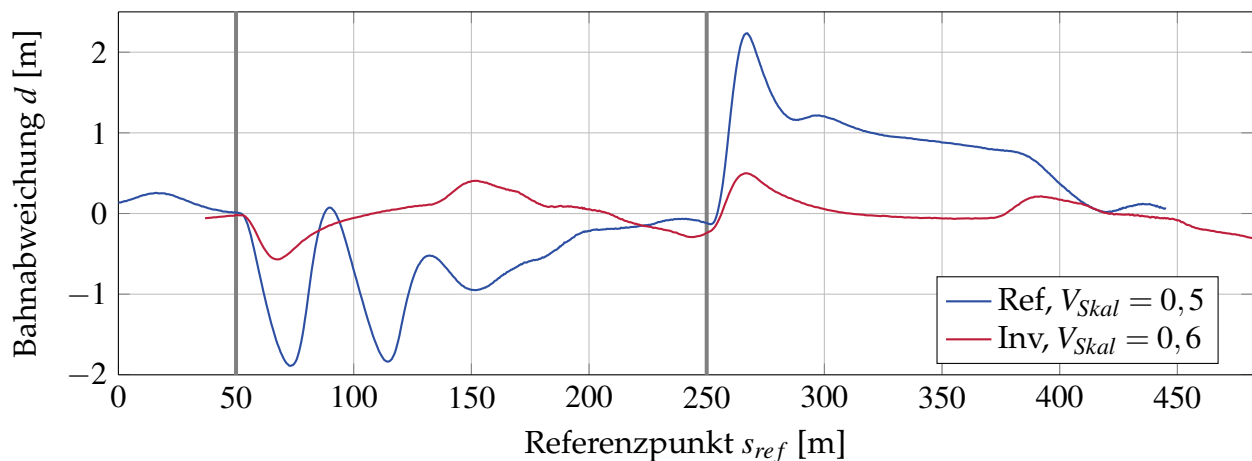


Abbildung 7.20: Vergleich der Bahnabweichungen des Referenz- und des Inversionsreglers für die automatische Fahrt mit Fehleraufschaltung

Im Vergleich zu der Fahrt mit dem entwickelten Inversionsregler kommt es bei Verwendung des Referenzreglers zu erheblichen Abweichungen nach Auf- und Abschaltung des Fehlers. So werden Bahnabweichungen von mehr als zwei Metern aufgebaut, die jeweils erst nach 150 Metern wieder ausgeglichen werden können. Aus diesem Ergebnis folgt, dass bei Verwendung des im Nominalfall hochperformanten Referenzreglers für den untersuchten Schadensfall keine zufriedenstellende Bahnführungsgenauigkeit umgesetzt werden kann. Die resultierenden Abweichungen können im tatsächlichen Straßenverkehr zu erheblichen

Gefährdungssituationen führen, was anschaulich den möglichen Sicherheitsgewinn durch den Einsatz adaptiver und lernfähiger Konzepte unterstreicht. So gelingt es dem entwickelten Regler den aufgebrachten Fehler schnell und stabil zu kompensieren und auf diese Weise eine sichere Fortführung der automatischen Fahrfunktion auch bei höheren Geschwindigkeiten zu gewährleisten. Anwendung findet ein solches Regelverhalten, wenn auch nach Eintritt und Kompensation eines kritischen Fehlers, ein automatischer Übergang in den risikominimalen Zustand sichergestellt werden soll.

7.3.2 Fehlerausgleich auf anspruchsvollem Rundkurs

Zum Abschluss der angestellten Untersuchungen wird der vorgestellte Fahrzeugregler auf einem, im Vergleich zum vorangegangenen Abschnitt, anspruchsvolleren Rundkurs bewertet. Neben einer Schikane, einer Haarnadelkurve und mehreren mit hohen Geschwindigkeiten zu durchfahrenden Kurvenabschnitten, besitzt die Strecke ein leichtes Höhenprofil, so dass es zu kurzen Bergauf- und Bergabpassagen kommt. Als Schadensfall wird erneut die Auf- und Abschaltung eines Lenkradsprungs verwendet. Aufgrund der für diesen Kurs existierenden Begrenzungen vorhandener Auslaufzonen abseits der Strecke und dem generell höherer Geschwindigkeitsniveau wird der Fehlerfall auf einen Lenkradsprung von 40° reduziert. Das vorgegebene Geschwindigkeitsprofil ist in Anhang C.7 abgebildet.

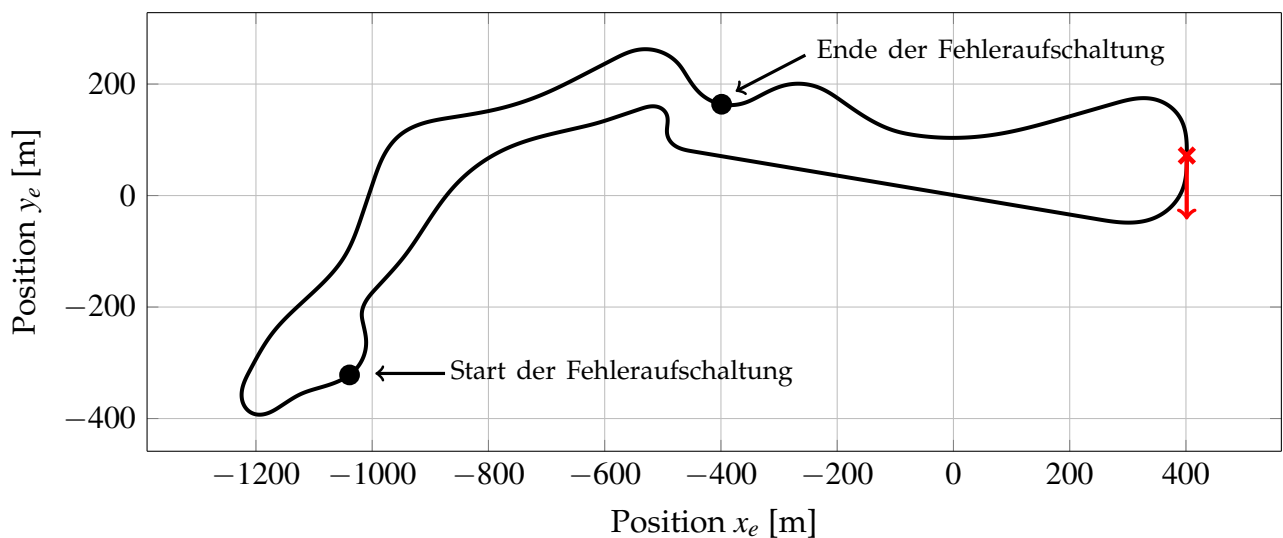


Abbildung 7.21: Anspruchsvoller Rundkurs für die Untersuchung des Fahrzeugreglers im Fahrversuch

Neben dem Ausgleich des aufgeschalteten Fehlers, ist es das Ziel der Untersuchungen, den in der Simulation gezeigten Einfluss von τ und λ in der Praxis zu bestätigen. So wurden, erneut unter der Maßgabe einer zunächst konservativen Inbetriebnahme, zwei Parametersätze für die automatische Fahrt mit dem Gleitzustandstraining untersucht. In Abbildung 7.22 sind die resultierenden Bahnabweichungen für das SMC-FO mit $\lambda = 40$ und $\tau = 0,3$, sowie einer zweiten Fahrt mit $\lambda = 80$ und $\tau = 0,6$ abgebildet. Man kann sehen, dass es auch für den

anspruchsvolleren Rundkurs für beide Fahrten zu einer stabilen und schnellen Kompensation des aufgeschalteten Fehlers kommt. Dabei ist für die zweite Fahrt mit verdoppelten Netzwerkparametern eine erhöhte Regelgüte erkennbar.

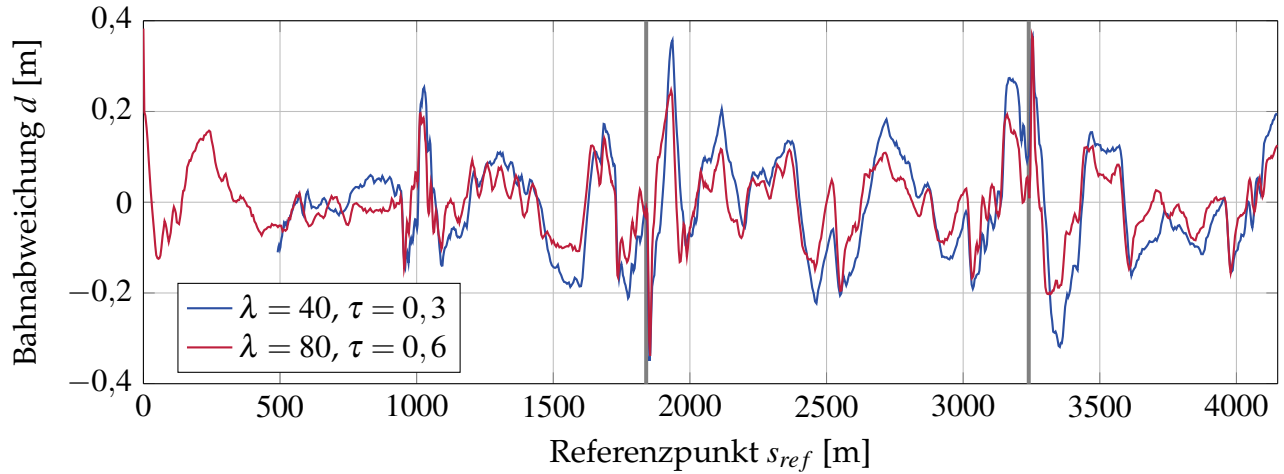


Abbildung 7.22: Vergleich der Bahnabweichungen für unterschiedliche Netzwerkparameter des Gleitzustandstrainings erster Ordnung

Mit Ausnahme des Moments der Fehlerschaltung, können so Bahnablagen von jederzeit weniger als 20cm erreicht werden. Somit kann das Ergebnis der Simulationen, dass es durch die Erhöhung von τ und λ zu einer Beschleunigung des Netzwerktrainings kommt, auch im Fahrversuch bestätigt werden. Der Einfluss der Parametererhöhung auf das Netzwerktraining ist in Abbildung 7.23 in Form der Frobeniusnorm dargestellt.

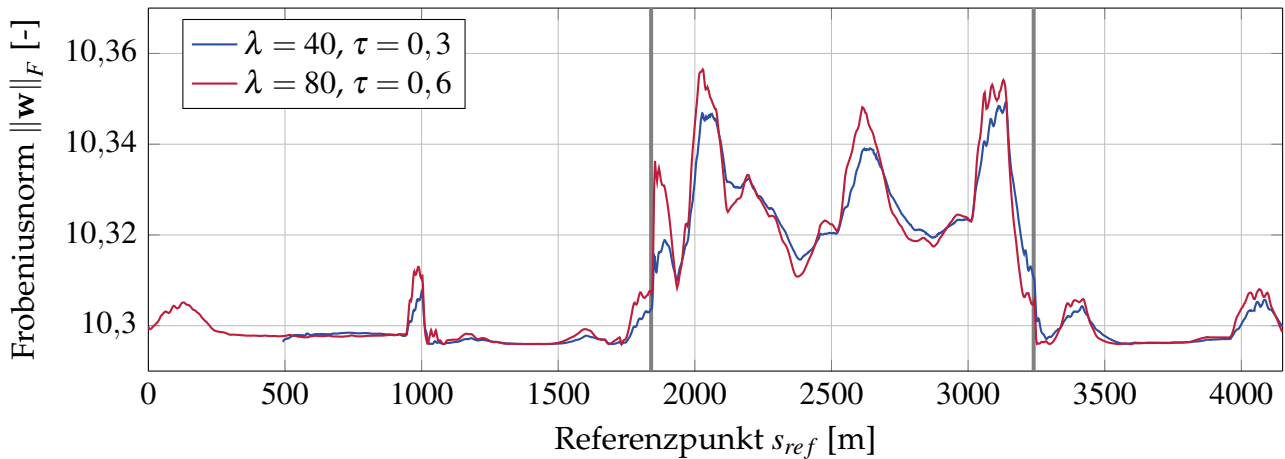


Abbildung 7.23: Vergleich der Frobeniusnorm für unterschiedliche Parameter des Gleitzustandstrainings erster Ordnung

Es ist zu erkennen, dass es dem Netzwerk für die Fahrt mit $\lambda = 80$ und $\tau = 0,6$ möglich ist, mit einem größeren Sprung der Netzwerknorm zu reagieren, was ein Indiz für die schnellere Anpassung der Netzwerkgewichte an das zu kompensierende Fehlerszenario ist. Darüber hinaus ist für die gesamte Zeit des Fehlerausgleichs eine höhere Normaktivität zu

beobachten, bevor es nach der Fehlerabschaltung zu einem mit der niedrigeren Parametrierung vergleichbaren Verlauf kommt. Es kann also auch für den Fahrversuch gezeigt werden, dass die Erhöhung von λ und τ keinen negativen Einfluss auf das Netzwerkverhalten im Nominalfall hat, im Moment einer Systembeschädigung jedoch zu einem beschleunigten Fehlerausgleich führt.

Die Fähigkeit des untersuchten Lernverfahrens auch mit konstanten Eingangsgrößen und der somit fehlenden Möglichkeit zum streckenabhängigen Wissensaufbau zu einer performanten Gierratenregelung zu führen, ist für die Fahrt mit $\lambda = 80$ und $\tau = 0,6$ in Abbildung 7.24 zu sehen. Trotz der bei der Umsetzung der Steuergrößenzuweisung getroffenen Vereinfachungen und den daraus resultierenden Inversionsfehlern bei der Reifenkraftberechnung, kann der kommandierten Gierratenvorgabe durch den Einsatz des neuronalen Netzwerks mit hoher Qualität gefolgt werden.

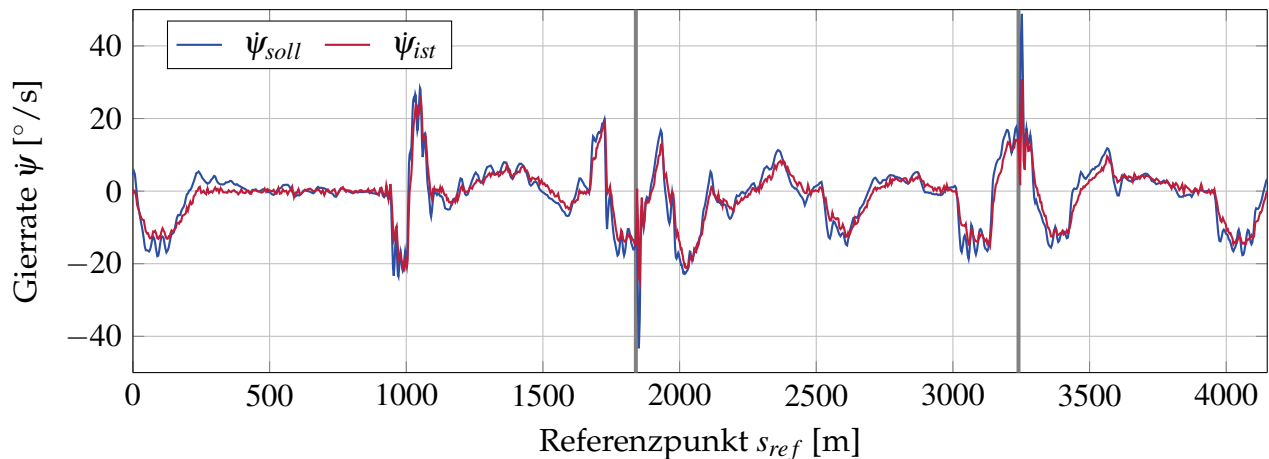


Abbildung 7.24: Vergleich der Gierratenregelung für das Gleitzustandstrainings erster Ordnung mit $\lambda = 80$ und $\tau = 0,6$

Dabei ist kein Unterschied zwischen der Regelgüte bei der Fahrt mit und ohne Fehlerabschaltung zu erkennen. Es kann somit abschließend gezeigt werden, dass der sprunghafte Ausgleich des aufgeschalteten Lenkwinkelfehlers auch im Fahrversuch keine Auswirkung auf die primäre Aufgabe des neuronalen Netzes in Form des kontinuierlichen Ausgleichs der bestehenden Inversionsfehler hat. Dies unterstreicht das Potential der Gleitzustandsverfahren, durch eine hohe Konvergenzgeschwindigkeit bei gleichzeitiger Robustheit infolge der dynamische Lernratenbestimmung, einen Beitrag zur Erhöhung der Leistungsfähigkeit von adaptiven Regelungsansätzen zu leisten.

Kapitel 8

Zusammenfassung und Ausblick

8.1 Zusammenfassung

In der vorliegenden Arbeit werden Lernverfahren neuronaler Netzwerke auf Basis der Gleitzustandsregelung erster und zweiter Ordnung entwickelt und ihre Anwendung in adaptiven Regelungsarchitekturen untersucht und bewertet. Dabei liegt der Fokus der Ausarbeitungen auf der systematischen Übertragung des Gleitzustandskonzeptes auf das Training neuronaler Netzwerke in Form der Herleitung eines definierbaren Schaltverhaltens mit garantierter Einhaltung abgeleiteter Stabilitätsbedingungen.

Den Ausgangspunkt der entwickelten Konzepte bilden die theoretischen Grundlagen der neuronalen Netzwerke. Dabei wird zunächst eine Verortung dieses Maschinenlernverfahrens im Forschungskontext der künstlichen Intelligenz vorgenommen. Im Anschluss wird der grundlegende Aufbau neuronaler Netze durch einzelne Neuronen zu einem verbundenen Netzwerk erläutert. Darüber hinaus wird als Grundlage für die spätere Erweiterung durch das Gleitzustandskonzept der klassische Backpropagation-Algorithmus hergeleitet und im Anschluss auf die Herausforderungen bei der Nutzung Online-trainierter Netzwerke in Regelungsanwendungen eingegangen. Die dargestellte Notwendigkeit detailliert beschreibbarer innerer Abläufe neuronaler Netzwerke wird dabei als Motivation für die spätere Anwendung der Gleitzustandsregelung auf das Netzwerktraining verwendet. Unter Nutzung der dargestellten Grundlagen wird die Wichtigkeit einer geeigneten Lernrate für ein stabiles und schnelles Netzwerktraining begründet und daraus der Beitrag der entwickelten Methoden für die Anwendungen Online-trainierter Netze in sicherheitskritischen Systemen abgeleitet.

Im Anschluss wird auf die Gleitzustandsregelung als Grundlage der in dieser Arbeit entwickelten Trainingsverfahren eingegangen. Dabei wird die Möglichkeit zur Vorgabe einer gewünschten Systemdynamik im Gleitzustand durch die Definition von Schaltgesetzen und -funktionen demonstriert und auf die Robustheit gegenüber externen Störungen eingegangen. Daran anschließend werden unterschiedliche Konzepte der Erreichbarkeits- und Existenzbedingung eines Gleitzustandes eingeführt. Mit der Differenzierung von

idealem und realem Schalten wird auf die besondere Herausforderung der Verwendung von Gleitzustandsverfahren erster Ordnung in zeitdiskreten Systemen eingegangen und so die Motivation für die Verwendung von SMC-Verfahren höherer Ordnung dargestellt.

Aufbauend auf der Erzeugung eines Gleitzustands erster Ordnung wird die Gleitordnung eines im Gleitzustand befindlichen Systems als Anzahl der benötigten Ableitungen der Schaltfunktion bis zum Auftauchen von Diskontinuitäten eingeführt und somit die Definition von Gleitzuständen höherer Ordnung gegeben. Der anschließend erläuterte Twisting-Algorithmus, mit diskontinuierlicher Umschaltung an der zweiten Ableitung der Schaltfunktion, dient als Grundlage für das später vorgestellte Gleitzustandstraining zweiter Ordnung.

Den Kern der Arbeit bilden die nun anschließenden Beschreibungen und Untersuchungen der Gleitzustandslernverfahren erster und zweiter Ordnung. Um den Transfer der zuvor vorgestellten Gleitzustandsregelung auf die Anwendung als Trainingsverfahren zu ermöglichen, wird zunächst der Paradigmenwechsel von einem optimierungsgetriebenen Prozess hin zu einem Regelungsprozess als Grundlage des Netzwerktrainings beschrieben. Auf diese Weise können der Netzwerkfehler und dessen Ableitungen als Systemzustände definiert werden, die durch das entwickelte Trainingsverfahren zunächst in den Gleitzustand und anschließend in die stabile Ruhelage überführt werden. Durch die Definition einer Schaltfunktion in Abhängigkeit des Netzwerkfehlers und einer Gewichtsänderungsvorschrift, die den konventionellen Backpropagation-Algorithmus um das Vorzeichen der Schaltfunktion ergänzt, wird die Basis für das Gleitzustandstraining erster und zweiter Ordnung geschaffen.

Unter Verwendung einer zeitdiskreten Erreichbarkeits- und Existenzbedingung wird eine Fallunterscheidung für die betragsmäßig Verkleinerung der Schaltfunktion in jedem Zeitschritt aufgestellt. Mit dieser wird es möglich, ein zustandsabhängiges Intervall für die Lernrate zu berechnen, dessen Einträge zum Erreichen des Gleitzustands und folglich zu einem stabilen Training führen. Ein interessantes Ergebnis stellt die Auswirkung der letztendlichen Lernratenwahl innerhalb des berechneten Intervalls dar. Mit Hilfe des in dieser Arbeit eingeführten Intervallparameters für die Wahl einer Lernrate innerhalb des aufgespannten Intervalls wird ein Konvergenzquotient gebildet, mit dem die Veränderung der Schaltfunktion im kommenden Zeitschritt angegeben werden kann. Darüber hinaus wird dargestellt, dass die angepasste Gewichtsänderungsvorschrift die Berechnung von negativen Lernraten zulässt, was als Kontrast zu konventionellen Backpropagation-Ansätzen steht.

Anhand zweier unterschiedlicher Trainingsszenarien wird gezeigt, dass die Netzwerkzustände durch das entwickelte Trainingsverfahren erster Ordnung stabil in einen Gleitzustand überführt werden und die durch den Konvergenzquotienten beschriebene Abhängigkeit vom Intervallparameter durch die Ergebnisse bestätigt werden kann. Somit

ist es für die gezeigten Trainingsbeispiele möglich, einen mit konventionellen Ansätzen in der Form bisher nicht zu realisierenden Grad an Determinismus bezüglich der Änderung des Ausgabefehlers im nächsten Zeitschritt zu erreichen. Als Basis für die spätere Parametrierung des Trainingsverfahrens für die Anwendung im geschlossenen Regelkreis, wird eine Studie zum Einfluss des Schaltfunktionsparameters auf das Trainingsergebnis durchgeführt. Dabei wird gezeigt, dass es innerhalb der aufgestellten Intervallgrenzen für jede Kombination von Intervall- und Schaltfunktionsparameter zu einem stabilen Training kommt und auf Basis des mittleren quadratischen Fehlers ein idealer Bereich für die Wahl der Netzwerkparameter angegeben werden kann.

Im Anschluss wird auf Basis des Twisting-Algorithmus das Gleitzustandstraining zweiter Ordnung eingeführt. Auf Grundlage einer erweiterten Kontraktionsbedingung für die betragsmäßigen Verkleinerung der ersten Schaltfunktionsableitung in jedem Zeitschritt, wird eine weitere Fallunterscheidung für die Berechnung eines zweiten Lernratenintervalls aufgestellt. Mit Hilfe eines zweiten Intervallparameters kann ein zusätzlicher Lernratenanteil berechnet werden, der die ursprünglich für das Verfahren erster Ordnung berechnete Rate erweitert. Dabei bleibt die Reduktion des Schaltfunktionsbetrags die maßgebliche Stabilitätsbedingung, so dass die kombinierte Lernrate weiterhin durch die Grenzen des Verfahrens erster Ordnung beschränkt bleibt.

Unter Verwendung der beiden bereits eingeführten Trainingsszenarien wird die Übertragbarkeit des Twisting-Algorithmus auf das Training neuronaler Netzwerke demonstriert, indem die Netzwerkzustände auf die für das Verfahren charakteristische wirbelartige Trajektorie in der Phasenebene gezwungen werden können. Nachdem der generelle Transfer des Ansatzes zweiter Ordnung auf das Netzwerktraining dargestellt ist, wird eine Parameterstudie zur gegenseitigen Beeinflussung der beiden Intervallparameter durchgeführt und dabei auf die Reduktion des Chatterings durch den Ansatz zweiter Ordnung eingegangen. Analog zum Vorgehen für das Verfahren erster Ordnung wird für die spätere Anwendung im geschlossenen Regelkreis gezeigt, dass alle untersuchten Parameterkombinationen zu einem stabilen Netzwerktraining führen und im Vergleich zum Gleitzustandsverfahren erster Ordnung vor allem die Charakteristik der Netzwerkausgabe bei gleichbleibender Trainingsgüte beeinflusst werden kann.

Nach der Vorstellung der entwickelten Trainingsverfahren wird die Theorie der nichtlinearen dynamischen Inversion als beispielhaftes Regelungskonzept für die Erweiterung um Online-trainierte neuronale Netzwerke dargestellt. Mit der Darstellung der Fehlerdynamik und ihrer Stabilisierung durch einen Linearregler sind die Grundlagen für die Erweiterung mit neuronalen Netzwerken geschaffen. Es wird erläutert, wie Inversionsfehler als Konsequenz von Parameter- und Modellunsicherheiten bei der Modellbildung sowie Systemdegradationen im laufenden Regelbetrieb durch den Einsatz neuronaler Netzwerke kompensiert werden können. Dabei wird insbesondere auf die Nachweise der Beschränktheit des Fehlervektors und der Netzwerkgewichte eingegangen, da diese,

durch die Wahl negativer Lernraten für das Gleitzustandstraining, gegenüber den aus der Literatur bekannten Ansätzen angepasst werden.

Als Möglichkeit zur simulativen Bewertung des entwickelten Flugreglers wird die verwendete Simulationsumgebung vorgestellt. Im Anschluss wird die Umsetzung der dynamischen Inversion als kaskadische Regelungsarchitektur mit drei Regelschleifen vom relativen Grad eins beschrieben. Für den Ausgleich vorhandener Inversionsfehler wird die innerste Reglerkaskade durch jeweils ein neuronales Netzwerk in der Roll-, Nick- und Gierratenregelung erweitert. Um die durch die innerste Kaskade kommandierten Momentenänderungen umsetzen zu können, wird eine Steuerflächenzuweisung vorgestellt, welche die benötigten Aktuatorausschläge der drei zur Verfügung stehenden Steuerflächen berechnet. Den Abschluss der Flugreglerumsetzung macht eine knappe Beschreibung des Vorgehens für die bandbreitenoptimale Wahl der Reglerparameter.

Zur Darstellung der generischen Natur der entwickelten Ansätze und als Möglichkeit zur Untersuchung der Leistungsfähigkeit in praktischen Versuchen wird im Anschluss eine zweite Umsetzung des vorgestellten Regelungskonzepts in Form eines Fahrzeugreglers vorgestellt. Dabei wird in übersichtlicher Form die verwendete Simulationsumgebung mit dynamischer Modellierung eines Zwei-Spur-Modells beschrieben. Anschließend wird die Umsetzung eines Querdynamikreglers auf Basis dynamischer Inversion, erneut mit Erweiterung um ein neuronales Netzwerk in der innersten Kaskade, vorgestellt. Um hochfrequente Lenkradbewegungen in der unmittelbaren Nähe der vorgegebenen Sollbahn zu reduzieren, wird eine Skalierung des Gierratenfehlers in Abhängigkeit der seitlichen Bahnabweichung eingeführt.

Die Bewertung des vorgestellten Regelungskonzeptes wird in drei Abschnitte unterteilt. Zu Beginn wird der Flugzeugregler in der vorgestellten Simulationsumgebung unter Einfluss von Sensorfehlermodellen sowie Wind und Turbulenz untersucht. Neben dem generellen Nachweis eines stabilen Regelungsergebnisses im schadensfreien Nominalfall, wird die Reaktion auf einen künstlich induzierten Inversionsfehler bewertet. Es wird deutlich, dass die Ergebnisse der im Vorfeld untersuchten Trainingsszenarien auf das adaptive Regelungskonzept eines kleinen unbemannten Flugsystems übertragen werden können. So wird gezeigt, dass alle untersuchten Intervallparameter zu einem stabilen Ausgleich des aufgeschalteten Fehlers führen und die Adaptionsgeschwindigkeit für Werte in Nähe der oberen Grenze zunimmt. Für das Verfahren zweiter Ordnung kann für große Intervallparameter darüber hinaus eine weitere, wenn auch nur noch geringe, Beschleunigung des Fehlerausgleichs gezeigt werden.

Zum Vergleich mit dem konventionellen Backpropagation-Algorithmus wird der Ausgleich einer lageabhängigen Schwerpunktverschiebung in Flugzeuglängsrichtung bei gleichzeitiger Aufbringung erheblicher Parameterunsicherheiten untersucht. Dabei kann die wesentliche Motivation gleitzustandsbasierter Trainingsverfahren bestätigt werden,

indem gezeigt wird, dass der Backpropagation-Algorithmus für die unterschiedlichen Fehlerfälle eine jeweils an das Szenario angepasste Lernrate benötigt. Es kann dabei jedoch nicht garantiert werden, dass es aufgrund einzelner kritischer Anregungen nicht doch zu Oszillationen kommt, die schlussendlich ein Versagen der Regelungsstrategie bewirken. Demgegenüber wird demonstriert, dass die entwickelten Gleitzustandsverfahren durch die ihnen zugrunde liegenden Erreichbarkeits- und Existenzbedingungen unabhängig von der Wahl der Trainingsparameter einen stabilen Ausgleich der unterschiedlichen Störungen ermöglichen. Dabei wird das Ergebnis der beispielhaft untersuchten Trainingsszenarien bestätigt, indem ein idealer Bereich für die Wahl der Netzwerkparameter existiert und eine darüber hinaus gehende Vergrößerung des Schaltfunktionsparameters zu immer geringeren Verbesserungen des mittleren quadratischen Fehlers führt. Es kann somit festgehalten werden, dass durch die dynamische und regelungstechnisch motivierte Berechnung einer stabilen Lernrate die Leistungsfähigkeit eines adaptiv gestützten Flugreglers auf Basis dynamischer Inversion auch in Gegenwart kritischer Systemschäden weiter erhöht werden kann.

Zur Demonstration der generischen Anwendbarkeit der entwickelten Gleitzustandsverfahren wird deren Langzeitstabilität und Leistungsfähigkeit in einer einstündigen Simulation des nichtlinearen Zwei-Spurmodells mit dem vorgestellten Querdynamikregler demonstriert. Anhand eines Rundkurses kann dabei die rundenweise Verbesserung der Regelgüte durch das kontinuierliche Netzwerktraining gezeigt werden. Bemerkenswert ist dabei, dass für die Parametrierung des Netzwerks erneut die im Flugregler untersuchten Werte für Schaltfunktions- und Intervallparameter verwendet werden können, ohne dass eine vorherige Anpassung an den Fahrzeugregler nötig wird. Zur weiteren Robustheitsuntersuchung der Lernverfahren erster und zweiter Ordnung gegenüber kritischen Systemdegradationen wird ein Platzen des vorderen linken Rades simuliert. Es zeigt sich, dass es, anders als für den Backpropagation-Algorithmus, nicht notwendig ist, das Netzwerk zum Ausgleich eines solchen Schadens umzuparametrieren, um so etwaigen Schwingungen vorzubeugen. Somit kann die charakteristische Robustheit von Gleitzustandsverfahren gegenüber externen Störungen auch in der Anwendung als Lernverfahren innerhalb eines adaptiven Regelungssystems bestätigt werden.

Zusammenfassend kann für die durchgeführten Simulationen festgestellt werden, dass alle untersuchten Kombinationen von Intervall- und Schaltfunktionsparametern sowie der auf das System einwirkenden Störungen für die Gleitzustandsverfahren erster und zweiter Ordnung zu stabilen Trainingsergebnissen von hoher Konvergenzgeschwindigkeit führen. Dabei kann sowohl für den Flug- als auch für den Fahrzeugregler der gleiche ideale Wertebereich für die neuen Netzwerkparameter verwendet werden. Somit kann der ansonsten oftmals langwierige, auf empirischen Voruntersuchungen beruhende und meist zu konservativen Ergebnissen führende Auslegungsprozess für neuronale Netzwerke in der Online-Anwendung deutlich vereinfacht werden. Zudem wird die Leistungsfähigkeit im Schadensfall durch die dynamisch an den aktuellen Systemzustand angepasste Lernratenbestimmung gegenüber konventionellen Verfahren erhöht.

Im dritten Teil der Auswertung werden Ergebnisse praktischer Fahrversuche auf zwei unterschiedlichen Rundkursen für das Gleitzustandstraining erster Ordnung dargestellt. Mit Hilfe eines aufgeschalteten Lenkradsprungs wird der vorgestellte adaptive Querdynamikregler mit einem hochperformanten Referenzregler im Schadensfall verglichen. Dabei kann gezeigt werden, dass durch das neuronale Netzwerk der aufgeschaltete Fehler für unterschiedliche Geschwindigkeitsvorgaben schnell und stabil kompensiert werden kann, wohingegen es für den Referenzregler auch für geringere Geschwindigkeiten zu deutlich größeren Bahnablagen kommt, die einen Abbruch der automatischen Fahrt notwendig machen. Für die Fahrt auf einem anspruchsvolleren Rundkurs wird im Anschluss der Einfluss von Schaltfunktions- und Intervallparameter auf den Fehlerausgleich untersucht. Dabei können die Ergebnisse aus der Simulation insoweit bestätigt werden, dass es durch die Erhöhung beider Werte zu einem schnelleren Ausgleich des aufgeschalteten Fehlers kommt, ohne dass daraus Nachteile für die fehlerfreie Fahrt resultieren. So wird gezeigt, dass die Güte der Gierratenregelung vor, während und nach der Fehleraufschaltung von vergleichbarer Qualität ist, was die Leistungsfähigkeit des entwickelten Lernverfahrens auch im praktischen Fahrversuch nochmals unterstreicht.

Abschließend kann festgestellt werden, dass es durch die auf Erreichbarkeits- und Existenzbedingungen basierende Berechnung eines stabilen Lernratenintervalls möglich ist, die aus der Theorie bekannte Robustheit von Gleitzustandsverfahren erster und zweiter Ordnung auf das Training neuronaler Netzwerke zu übertragen. Dabei erlaubt der Intervallfaktor eine Abschätzung der Netzwerkzustände im kommenden Zeitschritt und erhöht auf diese Weise die Transparenz Online-trainierter neuronaler Netzwerke in regelungstechnischen Anwendungen. Der erfolgreiche Ersteinsatz im praktischen Fahrversuch demonstriert die Tauglichkeit der entwickelten Verfahren, durch Kombination mit modellbasierten Regelungsverfahren ein adaptives Gesamtsystem zu bilden, dessen Leistungsfähigkeit und Potential zum stabilen Fehlerausgleich denen eines Standardverfahrens überlegen sind.

8.2 Ausblick und künftige Fragestellungen

Die in dieser Arbeit entwickelten Lernverfahren für neuronale Netzwerke in sicherheitskritischen Regelungsanwendungen und deren Untersuchungen in Theorie als auch in der Praxis werfen neue Ansatzpunkte für weitere wissenschaftliche Arbeiten auf. Dabei sind viele dieser künftigen Fragestellungen unabhängig von der geregelten Zielplattform zu betrachten, so dass sowohl fliegende als auch fahrende Systeme als Regelstrecke vorstellbar sind.

Als konsequente Weiterführung der bisher angestellten Untersuchungen gilt es, im nächsten Schritt die begonnen Arbeiten im Fahrversuch bezüglich der Erprobung des Gleitzustandstrainings zweiter Ordnung und allen voran der Verwendung dynamischer

Eingangsgrößen abzuschließen. Dabei sollten analog der simulativen Implementierung eine Eingangsgrößenskalierung, ein wohl parametrierter robustifizierender Term und ein aktiviertes Pseudo-Control Hedging umgesetzt werden. Auf dieser Grundlage können im Anschluss Aussagen über das Normwachstum als Folge wiederkehrender Fehleranregung, wie sie im Fall ohne Eingangsskalierung und PCH beobachtet wurden, getroffen und mit dem Verhalten in der Simulation verglichen werden.

Besonderes Interesse bei der Erforschung und Entwicklung von adaptiven Regelungskonzepten gilt der Überwachung der *Systemgesundheit* im laufenden Regelungsbetrieb. So existieren Ansätze, bei denen die Validierung von kritischen Systemkomponenten durch die garantierte Einhaltung definierter Leistungsindizes realisiert wird. Für diese Ansätze ist es somit notwendig zu jedem Zeitpunkt eine Aussage über die aktuelle Leistungsfähigkeit des adaptiven Gesamtsystems treffen zu können. Zu diesem Zweck muss eine geeignete Metrik gefunden werden, die auf Basis zur Verfügung stehender Systemzustände einen Leistungsindex berechnet. Im Hinblick auf die konkrete Überwachung neuronaler Netzwerke bietet sich hierfür augenscheinlich die Netzwerknorm an. Als Anhaltspunkt sowohl für die Änderungsgeschwindigkeit als auch das betragsmäßige Niveau aller Netzwerkgewichte können über die Norm direkte Rückschlüsse auf das Verhalten des neuronalen Netzes gezogen werden. Dabei gilt es zu untersuchen, in wie weit die Überwachung einzelner Netzwerkschichten einen noch höheren Informationsgrad bieten kann. Durch die Verwendung des Gleitzustandstrainings ist es ferner möglich, den aktuellen Wert der Schaltfunktion als maßgeblich von den Netzwerkzuständen abhängige Größe für die Berechnung einer Zustandsmetrik zu nutzen. Darüber hinaus können die Netzwerkausgabe und deren Charakteristik für die Bildung einer solchen überwachenden Metrik verwendet werden. Gelingt die dynamische Bewertung der Leistungsfähigkeit eines neuronalen Netzwerks im Online-Betrieb, kann auf diese Weise ein wichtiger Beitrag zur Einsatztauglichkeit neuro-adaptiver Regelungssysteme in sicherheitskritischen Anwendungen geleistet werden.

Ein weiterer Anknüpfungspunkt an die in dieser Arbeit vorgestellten Untersuchungen ist die Kombination der gleitzustandstrainierten Netzwerke mit andersartigen Netzwerktopologien und -konzepten. So können zunächst Shortcut- und Recurrent-Netzwerke auf eine Verbesserung durch die Verwendung eines Gleitzustandstrainings untersucht werden. Über diese klassischen Topologien hinausgehend, kann die erstmalige Anwendung von Gleitzustandsverfahren auf Convolutional-Netzwerken im Kontext des Deep-Learnings zu interessanten Ergebnissen führen. So bietet insbesondere das Feld der Deep-Driving-Ansätze als Kombination von Umfelderkennung und einer direkt daraus abgeleiteten Handlungsempfehlung das Potential durch neuartige Trainingsverfahren, weiter verbessert zu werden.

Literaturverzeichnis

- [1] BUNDESMINISTERIUM FÜR VERKEHR UND DIGITALE INFRASTRUKTUR: *Verordnung zur Regelung des Betriebs von unbemannten Fluggeräten*. Berlin, Januar 2017 <http://www.bmvi.de/SharedDocs/DE/Artikel/LR/151108-drohnen.html?nn=12830>. – Zugriff am 21.01.2017
- [2] ROJAS, R.: *Neural Networks - A Systematic Introduction*. Berlin : Springer-Verlag, 1996
- [3] HAYKIN, S.: *Neural Networks - A Comprehensive Foundation*. 2nd. Upper Saddle River, NJ : Prentice Hall, 1999
- [4] RUSSELL, S. ; NORVIG, P.: *Künstliche Intelligenz - Ein moderner Ansatz*. ISBN 3-8273-7089-2, München : Pearson Education, 2012
- [5] LUGER, G.F. ; STUBBLEFIELD, W. A.: *Artificial intelligence: structures and strategies for complex problem solving*. Benjamin/Cummings Pub. Co., 1993. – ISBN 9780805347807
- [6] NEGNEVITSKY, M.: *Artificial Intelligence: A Guide to Intelligent Systems*. Addison-Wesley, 2005
- [7] HORNIK, K. ; STINCHCOMBE, M. ; WHITE, H.: Multilayer Feedforward Networks are Universal Approximators. In: *Neural Networks 2* (1989), S. 359–366
- [8] FUNAHASHI, K. I.: On the Approximate Realization of Continuous Mappings by Neural Networks. In: *Neural Networks 2* (1989), S. 183–192
- [9] POGGIO, T. ; GIROSI, F.: Networks for approximation and learning. In: *Proceedings of the IEEE 78* (1990), Nr. 9, S. 1481–1497. – ISSN 0018–9219. – 10.1109/5.58326
- [10] NOWACK, J.: *Windkanal-Freiflugmessungen zur Bestimmung flugmechanischer Kenngrößen*, Fakultät für Maschinenwesen der Rheinisch-Westfälischen Technischen Hochschule Aachen, Diss., 2010
- [11] SEIFERT, J.: *Identifizierung nichtlinearer aerodynamischer Derivative mit einem Modularen Neuronalen Netzwerk*, Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, Institut für Systemdynamik und Flugmechanik, Diss., 2003
- [12] KOSMATOPOULOS, E. B. ; POLYCARPOU, M. M. ; CHRISTODOULOU, M. A. ; IOANNOU, P. A.: High-order neural network structures for identification of dynamical systems.

- In: *IEEE Transactions on Neural Networks* 6 (1995), Nr. 2, S. 422–431. – ISSN 1045–9227. – 10.1109/72.363477
- [13] NARENDRA, K. S. ; PARTHASARATHY, K.: Identification and control of dynamical systems using neural networks. In: *IEEE Transactions on Neural Networks* 1 (1990), Nr. 1, S. 4–27. – ISSN 1045–9227. – 10.1109/72.80202
- [14] TURNER, P. ; MONTAGUE, G. ; MORRIS, J.: Nonlinear and direction-dependent dynamic process modelling using neural networks. In: *IEE Proceedings - Control Theory and Applications* 143 (1996), Nr. 1, S. 44–48. – ISSN 1350–2379. – 10.1049/ip-cta:19960061
- [15] ZHOU, C. ; LIU, S. ; LIU, P.: Neural Network Pattern Recognition Based Non-intrusive Load Monitoring for a Residential Energy Management System. In: *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, 2016, S. 483–487. – 10.1109/ICISCE.2016.110
- [16] WALDEMARK, J. ; DOVNER, P. O. ; KARLSSON, J.: Hybrid neural network pattern recognition system for satellite measurements. In: *Neural Networks, 1995. Proceedings., IEEE International Conference on* Bd. 1, 1995, S. 195–199 vol.1. – 10.1109/ICNN.1995.488092
- [17] ZHENG, B. ; QIAN, W. ; CLARKE, L. P.: Multistage neural network for pattern recognition in mammogram screening. In: *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on* Bd. 6, 1994, S. 3437–3441 vol.6. – 10.1109/ICNN.1994.374887
- [18] WU, S. ; CHOW, T. W. S.: Induction machine fault detection using SOM-based RBF neural networks. In: *IEEE Transactions on Industrial Electronics* 51 (2004), Nr. 1, S. 183–194. – ISSN 0278–0046. – 10.1109/TIE.2003.821897
- [19] ZHANG, X. ; LIU, Y. ; RYSDYK, R. ; KWAN, C. ; XU, R.: An intelligent hierarchical approach to actuator fault diagnosis and accommodation. In: *Proc. IEEE Aerospace Conference*, 2006
- [20] KRÜGER, T. ; SCHNETTER, P. ; SACHS, F. ; VÖRSMANN, P.: A Combined Scheme of Nonlinear Adaptive Flight Control and Neural Network Fault Detection. In: *AIAA Infotech@Aerospace 2012*. Garden Grove, California, USA, June 19 - June 21 2012. – 10.2514/6.2012-2430
- [21] MAAS, W.: Networks of Spiking Neurons: The Third Generation of Neural Network Models. In: *Trans. Soc. Comput. Simul. Int.* 14 (1997), Dezember, Nr. 4, S. 1659–1671. – ISSN 0740–6797
- [22] ALNAJJAR, F. ; MURASE, K.: A Simple Aplysia-Like Spiking Neural Network to Generate Adaptive Behavior in Autonomous Robots. In: *Adaptive Behaviour* 16 (2008), Nr. 5, S. 306–324

- [23] VREEKEN, J.: Spiking neural networks, an introduction / Department of Information and Computing Sciences, Utrecht University. 2003 (UU-CS-2003-008). – Forschungsbericht
- [24] WALTER, F. ; RÖHRBEIN, F. ; KNOLL, A.: Neuromorphic implementations of neurobiological learning algorithms for spiking neural networks. In: *Neural Networks* 72 (2015), S. 152 – 167. – ISSN 0893–6080. – Neurobiologically Inspired Robotics: Enhanced Autonomy through Neuromorphic Cognition
- [25] SCHMIDHUBER, J.: Deep learning in neural networks: An overview. In: *Neural Networks* 61 (2015), S. 85 – 117. – ISSN 0893–6080. – [dx.doi.org/10.1016/j.neunet.2014.09.003](https://doi.org/10.1016/j.neunet.2014.09.003)
- [26] SZEGEDY, C. ; LIU, W. ; JIA, Y. ; SERMANET, P. ; REED, S. ; ANGUELOV, D. ; ERHAN, D. ; VANHOUCHE, V. ; RABINOVICH, A.: Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. – ISSN 1063–6919, S. 1–9. – [10.1109/CVPR.2015.7298594](https://doi.org/10.1109/CVPR.2015.7298594)
- [27] RAZAVIAN, A. S. ; AZIZPOUR, H. ; SULLIVAN, J. ; CARLSSON, S.: CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014. – ISSN 2160–7508, S. 512–519. – [10.1109/CVPRW.2014.131](https://doi.org/10.1109/CVPRW.2014.131)
- [28] SERMANET, P. ; LECUN, Y.: Traffic sign recognition with multi-scale Convolutional Networks. In: *The 2011 International Joint Conference on Neural Networks*, 2011. – ISSN 2161–4393, S. 2809–2813. – [10.1109/IJCNN.2011.6033589](https://doi.org/10.1109/IJCNN.2011.6033589)
- [29] GIRSHICK, R.: Fast R-CNN. In: *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, S. 1440–1448. – [10.1109/ICCV.2015.169](https://doi.org/10.1109/ICCV.2015.169)
- [30] LEE, C. S. ; WANG, M. H. ; YEN, S. J. ; WEI, T. H. ; WU, I. C. ; CHOU, P. C. ; CHOU, C. H. ; WANG, M. W. ; YAN, T. H.: Human vs. Computer Go: Review and Prospect [Discussion Forum]. In: *IEEE Computational Intelligence Magazine* 11 (2016), Aug, Nr. 3, S. 67–72. – ISSN 1556–603X. – [10.1109/MCI.2016.2572559](https://doi.org/10.1109/MCI.2016.2572559)
- [31] NATIONAL SCIENCE AND TECHNOLOGY COUNCIL, COMMITTEE ON TECHNOLOGY: *Preparing for the Future of Artificial Intelligence*. Washington, USA : Executive Office of the President, Oktober 2016 https://www.whitehouse.gov/sites/default/files/whitehouse_files/microsites/ostp/NSTC/preparing_for_the_future_of_ai.pdf. – Zugriff am 14.01.2017
- [32] CALISE, A. ; LEE, H. ; N.KIM: High Bandwidth Adaptive Flight Control. In: *AIAA Guidance, Navigation, and Control Conference*. Denver, CO, September 2000. – AIAA 2000-4551

- [33] RYSDYK, R. ; CALISE, A.: Robust Nonlinear Adaptive Flight Control for Consistent Handling Qualities. In: *IEEE Transactions on Control Systems Technology* 13 (2005), Nr. 6, S. 896–910
- [34] CALISE, A. J. ; RYSDYK, R. T.: Nonlinear adaptive flight control using neural networks. In: *IEEE Control Systems Magazine* 18 (1998), Nr. 6, S. 14–25. <http://dx.doi.org/10.1109/37.736008>. – DOI 10.1109/37.736008
- [35] CALISE, A. J.: Neural networks in nonlinear aircraft flight control. In: *IEEE Aerospace and Electronic Systems Magazine* 11 (1996), Nr. 7, S. 5–10. <http://dx.doi.org/10.1109/62.533965>. – DOI 10.1109/62.533965
- [36] JOHNSON, E. N. ; CALISE, A. ; CORBAN, J. E.: Adaptive Guidance and Control for Autonomous Launch Vehicles. In: *Proceedings of the IEEE Aerospace Conference*. Big Sky, Montana, USA, März 2001
- [37] KIM, N.: *Improved Methods in Neural Network-Based Adaptive Output Feedback Control with Applications to Flight Control*, School of Aerospace Engineering, Georgia Institute of Technology, Diss., 2003
- [38] ISIDORI, A.: *Nonlinear Control Systems*. Springer Verlag, Berlin, 1996
- [39] KHALIL, H.: *Nonlinear Systems*. Prentice Hall, New Jersey, 3rd ed., 2002
- [40] SLOTINE, J.J.E. ; LI, W.: *Applied Nonlinear Control*. Prentice Hall, 1991. – ISBN 9780130408907
- [41] KHEMAISSIA, S. ; MORRIS, A.S.: Neuro-adaptive control of robotic manipulators. In: *Robotica* 11 (1993), Nr. 5, S. 465–473. – 10.1017/S026357470001701X
- [42] PATHAK, K. ; FRANCH, J. ; AGRAWAL, S. K.: Velocity and position control of a wheeled inverted pendulum by partial feedback linearization. In: *IEEE Transactions on Robotics* 21 (2005), June, Nr. 3, S. 505–513. – ISSN 1552–3098
- [43] LUCA, A. de ; LUCIBELLO, P.: A general algorithm for dynamic feedback linearization of robots with elastic joints. In: *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)* Bd. 1, 1998. – ISSN 1050–4729, S. 504–510 vol.1. – 10.1109/ROBOT.1998.677024
- [44] BUGAJSKI, D. J. ; ENNS, D. F. ; ELGERSMA, M. R.: A Dynamic Inversion Based Control Law With Application to the High Angle of Attack Research Vehicle. In: *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 1990
- [45] SNELL, S. A. ; ENNS, D. F. ; GARRARD, W. L.: Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft. In: *AIAA Journal of Guidance, Navigation and Control* 15 (1992), Nr. 4, S. 976–984

- [46] HOLZAPFEL, F.: *Nichtlineare adaptive Regelung eines unbemannten Fluggerätes*, Lehrstuhl für Flugmechanik und Flugregelung, Technische Universität München, Diss., 2004
- [47] LORENZ, S.: *Adaptive Regelung zur Flugbereichserweiterung des Technologiedemonstrators ARTIS*, Technische Universität Braunschweig, Diss., 2010
- [48] KRÜGER, T.: *Zur Anwendung neuronaler Netzwerke in adaptiven Flugregelungssystemen*, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, Diss., 2012
- [49] CHOWDHARY, G. ; JOHNSON, E. N. ; CHANDRAMOHAN, R. ; KIMBRELL, M. S. ; CALISE, A.: Guidance and Control of Airplanes Under Actuator Failures and Severe Structural Damage. In: *Journal of Guidance, Control, and Dynamics* 36 (2013), S. 1093–1104
- [50] SCHOLZ, G. ; TROMMER, G. F.: Model based control of a quadrotor with tiltable rotors. In: *Gyroscopy and Navigation* 7 (2016), Nr. 1, S. 72–81. – ISSN 2075–1109. – 10.1134/S2075108716010120
- [51] PEDRO, J. O. ; PANDAY, A. ; DALA, L.: A Nonlinear Dynamic Inversion-Based Neurocontroller for Unmanned Combat Aerial Vehicles During Aerial Refuelling. In: *International Journal of Applied Mathematics and Computer Science* 23 (2013), Nr. 1, S. 75 – 90. – 10.2478/amcs-2013-0007
- [52] SCHUHMANN, J. ; LIU, Y.: *Applications of Neural Networks in High Assurance Systems*. ISBN 978-3-642-10689-7 : Springer Verlag, Berlin, 2010
- [53] JACKLIN, S. A.: Closing the Certification Gaps in Adaptive Flight Control Software. In: *Proceedings of AIAA Guidance, Navigation and Control Conference and Exhibit*. Honolulu, HI, USA, 2008. – 10.2514/6.2008-6988
- [54] BROWN, A.: *Nerve Cells and Nervous Systems*. Springer Verlag, Berlin, 1991
- [55] MCMULLOCH, W. S. ; PITTS, W. H.: A Logical Calculus of The Ideas Immanent in Nervous Activity. In: *Bulletin of Mathematical Biophysics* 5 (1943), S. 115–133
- [56] ROSENBLATT, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. In: *Psychological Review* 65 (1958), Nr. 6
- [57] BRYSON, A.E. ; HO, Y.C.: *Applied optimal control: optimization, estimation, and control*. Blaisdell Pub. Co., 1969 (Blaisdell book in the pure and applied sciences)
- [58] RUMELHART, D.E. ; MCCLELLAND: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, 1986 (Bradford books Bd. 1). – ISBN 9780262680530
- [59] AMUNTS, K. ; LINDNER, A. ; ZILLES, K.: The Human Brain Project: Neurowissenschaftliche Perspektiven und Beiträge aus Deutschland. In: *Neuroforum* 2/14 (2014), S. 222–229. – ISSN 0947–0875

- [60] LI, Wen-Ke ; HAUSKNECHT, Matthew J. ; STONE, Peter ; MAUK, Michael D.: Using a million cell simulation of the cerebellum: Network scaling and task generality. In: *Neural Networks* 47 (2013), S. 95–102
- [61] MARKRAM, H. u. a.: Reconstruction and Simulation of Neocortical Microcircuitry. In: *Cell* 163 (2015), Oktober, Nr. 2, S. 456–492. – ISSN 00928674. – 10.1016/j.cell.2015.09.029
- [62] REY, G. D. ; WENDER, K. F.: *Neuronale Netze: Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung*. Bern : Huber, 2011
- [63] KRIESEL, D.: *Ein kleiner Überblick über Neuronale Netze*. 2007 http://www.dkriesel.com/science/neural_networks. – Zugriff am 14.01.2017
- [64] FIESLER, E.: *Neural Network Topologies*. 1996
- [65] HEBB, Donald O.: *The Organization of Behavior*. New York : John Wiley, 1949
- [66] MÖSSNER, M.: *Optimierung einer adaptiven neuronalen Reglerstruktur und Stabilitätsanalyse der verwendeten Lernverfahren*, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, Studienarbeit, 2009
- [67] SCHUHMANN, J. ; GUPTA, P. ; LIU, Y. ; SCHUHMANN, J. (Hrsg.) ; LIU, Y. (Hrsg.): *Applications of Neural Networks in High Assurance Systems*. Application of Neural Networks in High Assurance Systems: A Survey. ISBN 978-3-642-10690-3 : Springer Verlag, Berlin Heidelberg, 2010. – S. 1 - 19
- [68] HEINERT, M.: Artificial neural networks - how to open the black boxes? In: *Application of Artificial Intelligence in Engineering Geodesy*. Wien, Österreich, 2008, S. 42–62. – ISBN 978-3-9501492-4-1
- [69] STEPANYAN, V. ; KRISHNAKUMAR, K. ; NGUYEN, N. ; EYKEREN, L. V.: Stability and Performance Metrics for Adaptive Flight Control. In: *Proceedings of AIAA Guidance, Navigation and Control Conference*. Chicago, IL, USA, 2009. – 10.2514/6.2009-5965
- [70] EMEL'YANOV, S. V.: Control of first order delay systems by means of an astatic controller and nonlinear corrections. In: *Automation and Remote Control* 8 (1959), S. 983–991
- [71] UTKIN, V.: Variable Structure Systems with Sliding Modes. In: *IEEE Transactions on Automatic Control* AC-22 (1977), Nr. 2, S. 212–222
- [72] UTKIN, V.: *Sliding Modes in Control and Optimization*. Springer Verlag, Berlin, 1992
- [73] YU, X. ; KAYNAK, O.: Sliding-Mode Control With Soft Computing: A Survey. In: *IEEE Transactions on Industrial Electronics* 56 (2009), Nr. 9, S. 3275–3285
- [74] UTKIN, V. ; YU, X. (Hrsg.) ; XU, J.-X. (Hrsg.): *Lecture Notes in Control and Information Sciences*. Bd. 274: *Variable Structure Systems: Towards the 21st Century*. First Stage of VSS: People and Events. Springer Verlag, Berlin, 2002

- [75] PISANO, A.: *Second Order Sliding Modes: Theory and Applications*, Universita Studi di Cagliari, Diss., 2000
- [76] LEVANT, A.: *Introduction to Higher-Order Sliding Modes*. Tel-Aviv University, 2002-2003
- [77] FRIDMAN, L. ; LEVANT, A. ; PERRUQUETTI, W. (Hrsg.) ; BARBOT, J. P. (Hrsg.): *Higher Order Sliding Modes*. Sliding Mode Control in Engineering. Bertrams, 2002 (Control Engineering Series)
- [78] FILIPPOV, A. F. ; ARSCOTT, F. M.: *Differential equations with discontinuous righthand sides*. Dordrecht : Kluwer Academic Publ, 1988 (Mathematics and its applications. Soviet series). – ISBN 90-277-2699-X
- [79] POLYAKOV, A. ; FRIDMAN, L.: Stability notions and Lyapunov functions for sliding mode control systems. In: *Journal of the Franklin Institute* 351 (2014), Nr. 4, S. 1831 – 1865. ISBN <http://dx.doi.org/10.1016/j.jfranklin.2014.01.002>. – Special Issue on 2010-2012 Advances in Variable Structure Systems and Sliding Mode Algorithms
- [80] HEBISCH, H.: *Grundlagen der Sliding-Mode-Regelung*. Univ.-GH, Meß-, Steuer- und Regelungstechnik, 1995 (Forschungsbericht / Meß-, Steuer- und Regelungstechnik)
- [81] LASALLE, J. ; LEFSCHETZ, S.: *Die Stabilitätstheorie von Ljapunow*. Bibliographisches Institut, Mannheim, 1967
- [82] YOUNG, K.-K. D.: Controller design for a manipulator using theory of variable structure systems. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-8 (1978), Nr. 2, S. 101–109
- [83] UTKIN, V. ; GULDNER, J. ; SHI, J.: *Sliding Mode Control in Electro-Mechanical Systems*. CRC Press, London, 2009
- [84] HUNG, J. ; GAO, W. ; HUNG, J.: Variable Structure Control: A Survey. In: *IEEE Transactions on Industrial Electronics* 40 (1993), Nr. 1, S. 2–22
- [85] GAO, W. ; HUNG, J.: Variable Structure Control of Nonlinear Systems: A New Approach. In: *IEEE Transactions on Industrial Electronics* 40 (1993), Nr. 1, S. 45–55
- [86] ITKIS, U.: *Control systems of variable structure*. John Wiley & Sons, Incorporated, 1976 (A Halsted Press book). – ISBN 9780470150726
- [87] MARCINEK, M.: *Anwendung strukturvariabler Regelung auf Trainingsverfahren neuronaler Netze in der Flugregelung*, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, Studienarbeit, 2012
- [88] KASTE, J.: *Entwicklung und Bewertung gleitzustandsbasierter Lernverfahren höherer Ordnung für neuronale Netzwerke*, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, Studienarbeit, 2013

- [89] SCHNETTER, P. ; KASTE, J. ; KRÜGER, T.: Gleitzustandstraining Höherer Ordnung für die Adaptive Flugregelung eines Unbemannten Flugsystems. In: *Deutscher Luft- und Raumfahrtkongress*. Stuttgart, 10. - 12. September 2013
- [90] SCHNETTER, P. ; KASTE, J. ; KRÜGER, T.: Advanced Sliding Mode Online Training for Neural Network Flight Control Applications. In: *Proceedings of AIAA Guidance, Navigation, and Control Conference 2015*. Kissimmee, Florida, USA, January 05 - January 09 2015
- [91] CHAIREZ, I. ; POZNYAK, A. ; POZNYAK, T. ; BARTOLINI, G. (Hrsg.) ; FRIDMAN, L. (Hrsg.) ; PISANO, A. (Hrsg.) ; USAI, E. (Hrsg.): *Modern Sliding Mode Theorie - New Perspectives and Applications*. Springer Verlag, 2008 (High Order Sliding Mode Neuro-control for Uncertain Nonlinear SISO Systems: Theory and Application)
- [92] SCHNETTER, P. ; MARCINEK, M. ; KRÜGER, T. ; VÖRSMANN, P.: Adaptive Flight Control Using Second Order Sliding Mode Online Learning. In: *Proceedings of AIAA Infotech@Aerospace 2013*. Boston, Massachusetts, USA, August 19 - August 22 2013. – 10.2514/6.2013-5133
- [93] LEVANT, A.: Universal Output-Feedback SISO Controllers. In: *Asian Journal of Control* 5 (2003), Nr. 4, S. 484–497
- [94] FRIDMAN, L. ; MORENO, J. A. ; BANDYOPADHYAY, B. ; KAMAL, S. ; CHALANGA, A. ; YU, X. (Hrsg.) ; EFE, M. Ö. (Hrsg.): *Recent Advances in Sliding Modes: From Control to Intelligent Mechatronics*. Continuous Nested Algorithms : The Fifth Generation of Sliding Mode Controllers. Cham : Springer International Publishing, 2015. – 5–35 S. – ISBN 978-3-319-18290-2
- [95] TRIVEDI, P.: A Simple Reaching Law Based Design Method for 2-Sliding Mode Control. In: *Proc. IEEE International Conference on Industrial Technology*, 2010
- [96] KAYNAK, O. ; ERBATUR, K. ; ERTUGRUL, M.: The Fusion of Computationally Intelligent Methodologies and Sliding-Mode Control - A Survey. In: *IEEE Transactions on Industrial Electronics* 48 (2001), Nr. 1, S. 4–17
- [97] ZHANG, H. ; DU, M. ; BU, W.: Sliding Mode Controller with RBF Neural Network for Manipulator Trajectory Tracking. In: *IAENG International Journal of Applied Mathematics* 45 (2015), Nr. 4, S. 334–342
- [98] SIRA-RAMIREZ, H. ; COLINA-MORLES, E.: A Sliding Mode Strategy for Adaptive Learning in Adalines. In: *IEEE Transactions on Circuits and Systems - I Fundamental Theory and Applications* 42 (1995), Nr. 12, S. 2687–2691
- [99] PARMA, G. G. ; MENEZES, B. R. ; BRAGA, A. P.: Sliding Mode Algorithm for Training Multilayer Artificial Neural Networks. In: *IEEE Electronics Letters* 34 (1998), Nr. 1, S. 97–98

- [100] EFE, M. Ö. ; KAYNAK, O.: Stabilizing and robustifying the learning mechanisms of artificial neural networks in control engineering applications. In: *Int. J. Intell. Syst.* 15 (2000), Nr. 5, S. 365–388
- [101] NIED, A. ; SELEME, S. I. ; PARMA, G. G. ; MENEZES, B. R.: On-line neural training algorithm with sliding mode control and adaptive learning rate. In: *Neurocomputing* 70 (2007), S. 2687–2691
- [102] KRÜGER, T. ; SCHNETTER, P. ; PLACZEK, R. ; VÖRSMANN, P.: Fault-Tolerant Nonlinear Adaptive Flight Control Using Sliding Mode Online Learning. In: *Neural Networks* (2012). <http://dx.doi.org/10.1016/j.neunet.2012.02.025>.. – DOI 10.1016/j.neunet.2012.02.025.
- [103] PARMA, G. G. ; MENEZES, B. R. ; BRAGA, A. P.: Sliding mode backpropagation: control theory applied to neural network learnnig. In: *Proc. Int. Joint Conf. Neural Networks IJCNN '99*. Washington, DC, USA, 10 Jul - 16 Jul 1999, S. 1774 – 1778 vol.3. – ISBN:0-7803-5529-6
- [104] SCHECK, M.: *Anwendung von stabilen Sliding-Mode Trainingsverfahren auf neuronale Regler*, Institut für Luft- und Raumfahrtsysteme, Technische Universität Braunschweig, Studienarbeit, 2011
- [105] SARPTURK, S. ; ISTEKANOPULOS, Y. ; KAYNAK, O.: On the Stability of Discrete-Time Sliding Mode Control Systems. In: *IEEE Transactions on Automatic Control* AC-32 (1987), October, Nr. 10, S. 930–932
- [106] CHANG, K.-M. ; ZHU, Z.-H.: Discrete-Time Sliding Mode Controller Design with Weak Pseudo Sliding Condition. In: *Journal of Mathematical Analysis and Applications* 258 (2001), Nr. 2, S. 536 – 555. – <http://dx.doi.org/10.1006/jmaa.2000.7387>
- [107] TOPALOV, A. V. ; KAYNAK, O.: Online Learning in Adaptive Neurocontrol Schemes with a Sliding Mode Algorithm. In: *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics* 31 (2001), Nr. 3, S. 445–450
- [108] NIED, A. ; OLIVEIRA, J. de ; BARTOSZEWICZ, A. (Hrsg.): *Sliding Mode Control. Sliding Mode Control Approach for Training On-line Neural Networks with Adaptive Learning Rate*. Intech, Kroatien, 2011. – DOI: 10.5772/632
- [109] LIN, C.-M. ; TAI, C.-F. ; CHUNG, C.-C.: Intelligent control system design for UAV using a recurrent wavelet neural network. In: *Neural Computing and Applications* 24 (2014), Nr. 2, S. 487–496. – 10.1007/s00521-012-1242-5
- [110] LEWIS, F. L. ; YEGILDIREK, A. ; LIU, K.: Multilayer neural-net robot controller with guaranteed tracking performance. In: *IEEE Transactions on Neural Networks* 7 (1996), Nr. 2, S. 388–399. <http://dx.doi.org/10.1109/72.485674>. – DOI 10.1109/72.485674

- [111] SUN, X.D. ; CLARKE, T.: Advanced aircraft flight control using nonlinear inverse dynamics. In: *IEE - Proceedings - Control Theory and Applications* Bd. 141, 1994, S. 418 – 426. – 10.1049/ip-cta:19941580
- [112] SMITH, G. A. ; MEYER, G.: Aircraft Automatic Flight Control System with Model Inversion. In: *AIAA Journal of Guidance, Navigation and Control* 10 (1987), Nr. 3, S. 269–275
- [113] LOOYE, G. ; JOOS, H.-D.: Design of Robust Dynamic Inversion Control Laws using Multi-Objective Optimization. In: *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*. Montreal, Kanada, 6 - 9 August 2001
- [114] REINER, J. ; BALAS, G. K. ; GARRARD, W. L.: Robust Dynamic Inversion for Control of Highly Maneuverable Aircraft. In: *AIAA Journal of Guidance, Control, and Dynamics* 18 (1995), Nr. 1, S. 18–24. – 10.2514/3.56651
- [115] MILLER, C. J.: Nonlinear Dynamic Inversion Baseline Control Law: Flight-Test Results for the Full-scale Advanced Systems Testbed F/A-18 Airplane. In: *Proceedings of AIAA Guidance, Navigation, and Control Conference*. Portland, Oregon, USA, 08 - 11 August 2011. – 10.2514/6.2011-6468
- [116] JOHNSON, E. N.: *Limited Authority Adaptive Flight Control*, Georgia Institute of Technology, Diss., 2000
- [117] RYSDYK, R. ; AGARWAL, R. K.: Nonlinear Adaptive Flight Path and Speed Control using Energy Principles. In: *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*. Monterey, CA, USA, 05 - 08 August 2002
- [118] SACHS, F.: *Analyse der Rekonfigurationseigenschaften eines Flugreglers auf Basis dynamischer Inversion*, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, Studienarbeit, 2011
- [119] GROTH, T.: *Umsetzung und Optimierung einer fehlertoleranten und adaptiven Flugregelungsarchitektur auf einem Bewegungssimulator*, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, Masterarbeit, 2014
- [120] SCHNETTER, P. ; GROTH, T. ; KRÜGER, T.: Simulatorversuche eines Adaptiven und Rekonfigurierbaren Flugreglers zur Kompensation von Systemschäden. In: *Deutscher Luft- und Raumfahrtkongress*. Augsburg, 16. - 18. September 2014
- [121] SCHUMANN, J. ; NELSON, S.: Torward V & V of Neural Network Based Controllers. In: *Workshop on Self-Healing Systems, WOSS 2002*. Charleston, SC, USA, 2002, S. 745 – 755
- [122] HEISE, C. D. ; LEITÃO, M. ; HOLZAPFEL, F.: Performance and Robustness Metrics for Adaptive Flight Control - Available Approaches. In: *Proceedings of AIAA Guidance,*

- Naviagtion, and Control Conference*. Boston, Massachusetts, USA, August 19 - August 22 2013. – 10.2514/6.2013-5090
- [123] HÖCHT, L.: *Advances in Stability Analysis for Model Reference Adaptive Control Systems and Application to Unmanned Aerial Systems*, Lehrstuhl für Flugsystemdynamik, Technische Universität München, Diss., 2014
- [124] SVARICEK, F.: *Nullodynamik linearer und nichtlinearer Systeme: Definition, Eigenschaften und Anwendungen*, Institut für Systemdynamik und Flugmechanik, UniBW München, Aufsatz, 2006
- [125] UNBEHAUEN, H.: *Regelungstechnik I - Klassische Verfahren zur Analyse und Synthese linearer kontinuierlicher Regelsysteme*. Vieweg Verlag, Braunschweig, 1992. – ISBN 3-528-63332-8
- [126] SCHNETTER, P.: *Entwicklung eines Flugreglers auf Basis dynamischer Inversion*, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, Diplomarbeit, 2011
- [127] BROCKHAUS, R. ; ALLES, W. ; LUCKNER, R.: *Flugregelung*. ISBN 978-3-642-01442-0 : Springer Verlag, Berlin, 2011
- [128] SCHOLZ, M.: *Untersuchung der Flugeigenschaften des unbemannten Forschungsflugzeugs "ALADINA"*, Institut für Flugführung, Technische Universität Braunschweig, Masterarbeit, 2015
- [129] KRÜGER, T.: *Erweitertes Kalman-Filter zur Sensordatenfusion in Navigationsanwendungen*, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, Diplomarbeit, 2011
- [130] WILKENS, C.-S. ; MARTIN, T. ; SCHOLTZ, A. ; VÖRSMANN, P. ; REINHOLD, M. ; SELSAM, P.: Project ANDROMEDA: Automated Generation of Aerial Photographs with Unmanned Aircraft Systems. In: *European Journal of Navigation* 8 (2010), Nr. 3, S. 32 – 38
- [131] WINKLER, S.: *Zur Sensordatenfusion für integrierte Navigationssysteme unbemannter Kleinstflugzeuge*, Zentrum für Luft- und Raumfahrt, Technische Universität Braunschweig, Diss., 2007
- [132] SCHULZ, H.-W.: *Aufbau und Durchführung von Versuchen zur Vermessung von Modellbau-Servos und Ermittlung theoretischer Modelle zur Beschreibung des dynamischen Übertragungsverhaltens*, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, Diplomarbeit, 2002
- [133] SCHOLZ, G.: *Modellfreie adaptive Flugregelung mittels neuronaler Netzwerke*. Studienarbeit, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, 2011

- [134] DRYDEN, H.L. ; FRIEDLANDER, S.K. (Hrsg.) ; TOPPER, L. (Hrsg.): *A Review of the Statistical Theory of Turbulence*. Turbulence - Classic Papers on Statistical Theory. Interscience Publishers, Inc., New York, 1961. – Nachdruck aus Quart. Appl. Math. I, S. 7 - 42, 1943
- [135] BOSSDORF-ZIMMER, B.: *Nichtlineare Fahrzustandsbeobachtung für die Echtzeitanwendung*. Shaker, 2007 (Schriftenreihe des Instituts für Fahrzeugtechnik, TU Braunschweig). – ISBN 9783832263836
- [136] KÜÇÜKAY, F. ; FRÖMMIG, L.: Fahrzeugtechnik 3/ Fahrdynamik. In: *Skript zur Vorlesung. Institut für Fahrzeugtechnik, Technische Universität Braunschweig* (2012)
- [137] PACEJKA, H. B. ; BESSELINK, I. J. M.: Magic Formula Tyre Model with Transient Properties. In: *Vehicle System Dynamics* 27 (1997), Nr. 001, S. 234–249. – 10.1080/00423119708969658
- [138] AMMON, D.: *Modellbildung und Systementwicklung in der Fahrzeugdynamik*. Stuttgart : Teubner, 1997 (Leitfäden der angewandten Mathematik und Mechanik). – Univ., Habil.-Schr., Karlsruhe
- [139] LAUMANN, N.: *Integrale Reglerstruktur zur effektiven Abstimmung von Fahrdynamikssystemen*, Institut für Kraftfahrwesen, RWTH Aachen, Diss., 2007
- [140] HO, M.L. ; CHAN, P.T. ; RAD, A.B. ; SHIRAZI, M. ; CINA, M.: A novel fused neural network controller for lateral control of autonomous vehicles. In: *Applied Soft Computing* 12 (2012), Nr. 11, S. 3514 – 3525. – ISSN 1568–4946. – <http://dx.doi.org/10.1016/j.asoc.2012.07.004>
- [141] KASTE, J.: *Bewertung künstlicher neuronaler Netzwerke als robustes Bahnregelungskonzept in nichtlinearen und störungsbehafteten Fahrzuständen*, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, Diplomarbeit, 2014
- [142] BLYTHE, W. ; DAY, T.D. ; GRIMES, W.D.: 3-Dimensional Simulation of Vehicle Response to Tire Blow-outs. In: *SAE International* (1998), Nr. 980221

Anhang A

Abbildungen zu den Gleitzustandsverfahren

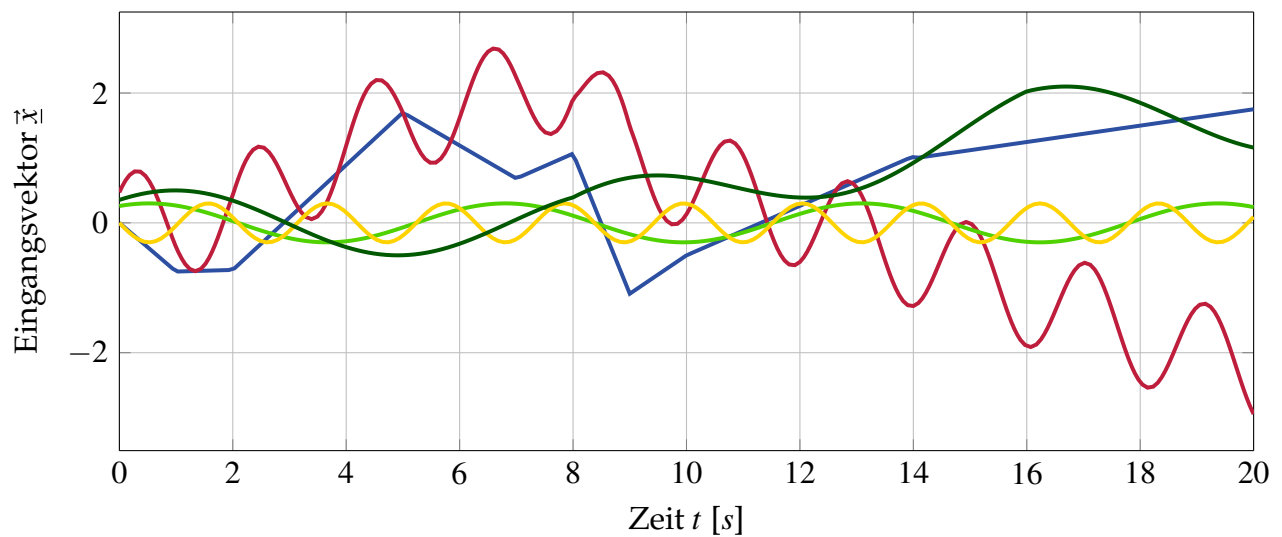


Abbildung A.1: Einträge des Eingabevektors der trainierten neuronalen Netzwerke für das komplexe Trainingsszenario

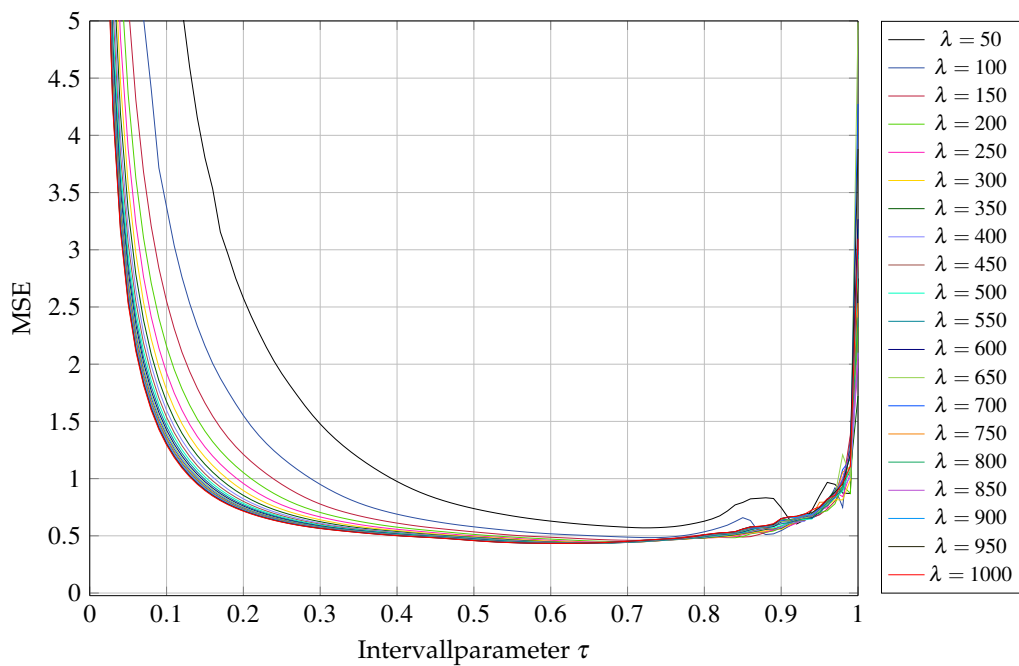


Abbildung A.2: MSE Vergleich für verschiedene λ in Abhängigkeit des Intervallparameters τ für das Netzwerktraining ohne σ^* -Korrektur

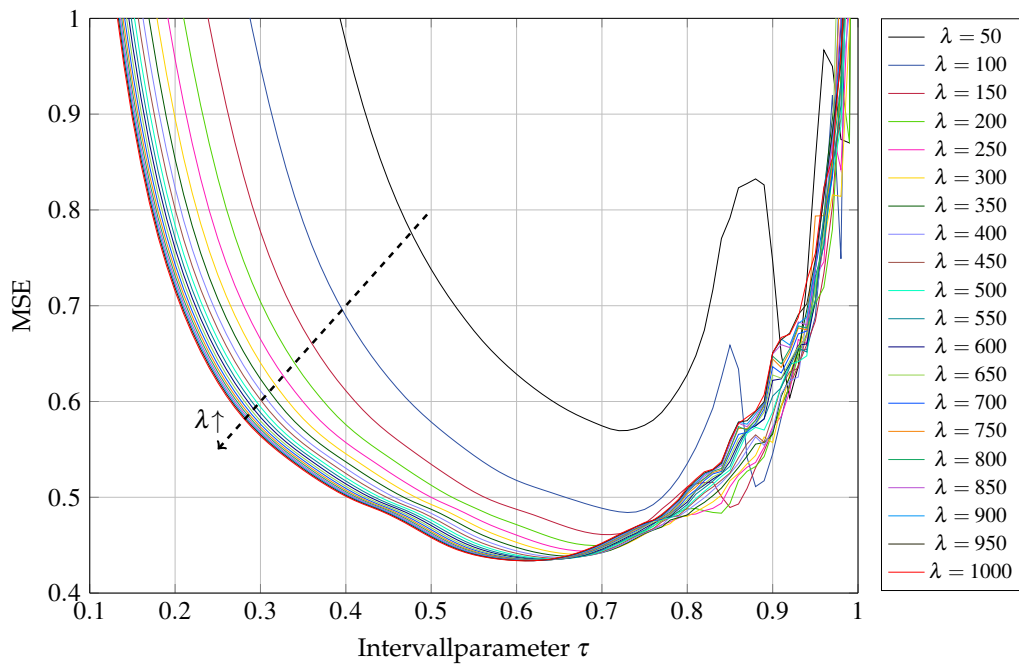


Abbildung A.3: Vergrößerter Ausschnitt des MSE Vergleichs für verschiedene λ in Abhängigkeit des Intervallparameters τ für das Netzwerktraining ohne σ^* -Korrektur

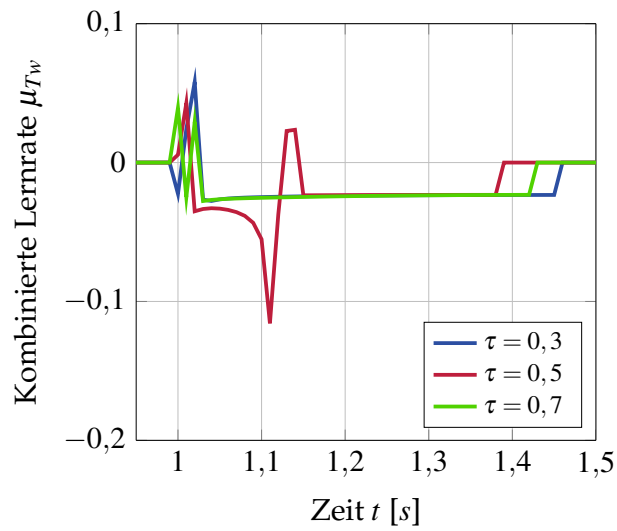


Abbildung A.4: Kombinierte Lernrate für verschiedene Intervallparameter bei einem Sprung der Zielgröße mit $\tau = 0,7$

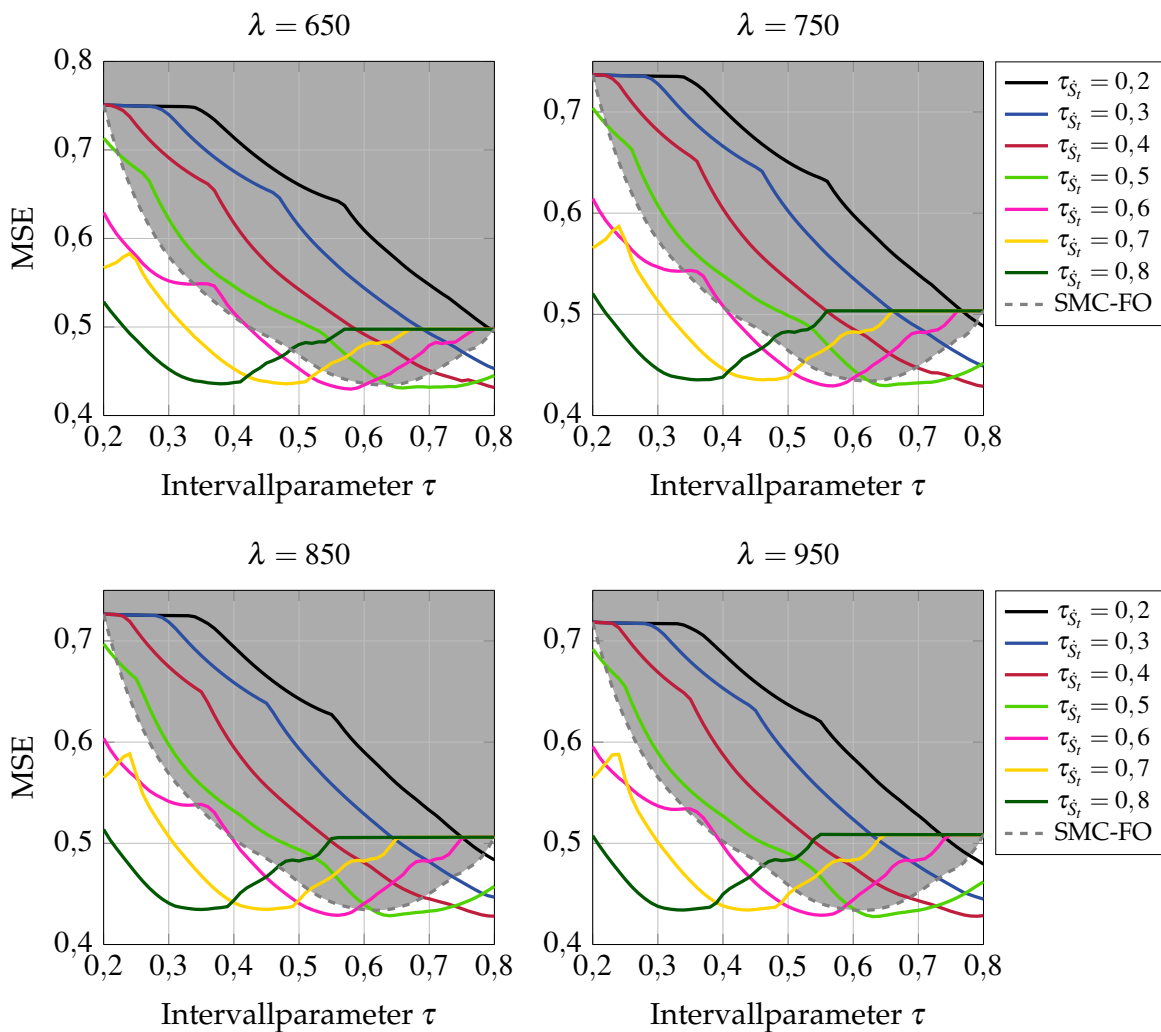


Abbildung A.5: MSE Vergleich in Abhängigkeit von τ , τ_s und λ für $\lambda = [650...950]$

Anhang B

Erweiterung der Beschränktheitsuntersuchung

Obere Schranke der Gewichtsmatrix

Die auch weiterhin für das Training mit dem Gleitzustandsverfahren geltende Beschränktheit der Netzwerkgewichte kann, entsprechend der in [46] beschriebenen Bestimmung der oberen Gewichtsgrenze des klassischen Backpropagations-Algorithmus, auf gleiche Weise für die in dieser Arbeit untersuchten Verfahren demonstriert werden. Zu diesem Zweck wird die positiv definite Ljapunow-Kandidatenfunktion

$$V = \frac{1}{2} \text{Spur} \left\{ \mathbf{w}^{(1)T} |\mu^{(1)}|^{-1} \mathbf{w}^{(1)} \right\} + \frac{1}{2} \text{Spur} \left\{ \mathbf{w}^{(2)T} |\mu^{(2)}|^{-1} \mathbf{w}^{(2)} \right\} \quad (\text{B.1})$$

aufgestellt und für den Beweis $\dot{V} < 0$ die entsprechend Ableitung der Form

$$\dot{V} = \text{Spur} \left\{ \mathbf{w}^{(1)T} |\mu^{(1)}|^{-1} \dot{\mathbf{w}}^{(1)} \right\} + \text{Spur} \left\{ \mathbf{w}^{(2)T} |\mu^{(2)}|^{-1} \dot{\mathbf{w}}^{(2)} \right\} \quad (\text{B.2})$$

gebildet. Im Anschluss wird die erweiterten Gewichtsänderungen aus den Gleichungen 5.59 und 5.60 eingesetzt und der jeweilige Lernratenbetrag heraus gekürzt, so dass ausschließlich das Vorzeichen der Lernrate in der Gleichung stehen bleibt:

$$\begin{aligned} \dot{V} = & \text{Spur} \left\{ \mathbf{w}^{(1)T} \cdot \text{sign}(\mu^{(1)}) \cdot \text{sign}(S_t) \cdot \left[\vec{x} \cdot |\vec{\zeta}| \cdot \mathbf{w}^{(2)T} \cdot \mathbf{f}' - \lambda_* \cdot \text{sign}(\vec{\zeta}) \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(1)} \right] \right\} \\ & + \text{Spur} \left\{ \mathbf{w}^{(2)T} \cdot \text{sign}(\mu^{(2)}) \cdot \text{sign}(S_t) \cdot \left[\left(\vec{f}(\mathbf{w}^{(1)} \vec{x}) - \mathbf{f}' \cdot \mathbf{w}^{(1)T} \vec{x} \right) \cdot |\vec{\zeta}| \right. \right. \\ & \left. \left. - \lambda_* \cdot \text{sign}(\vec{\zeta}) \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(2)} \right] \right\}. \end{aligned} \quad (\text{B.3})$$

Weiteres Umformen und Zusammenfassen unter dem Wissen, dass $\mu^{(1)} = \mu^{(2)}$ gilt, ergibt im Anschluss:

$$\begin{aligned} \dot{V} = & \text{Spur} \left\{ \mathbf{w}^{(1)T} \cdot \text{sign}(\mu^{(1)}) \cdot \text{sign}(S_t) \cdot \vec{x} \cdot |\vec{\zeta}| \cdot \mathbf{w}^{(2)T} \cdot \mathbf{f}' \right\} \\ & + \text{Spur} \left\{ \mathbf{w}^{(2)T} \cdot \text{sign}(\mu^{(2)}) \cdot \text{sign}(S_t) \cdot \left(\vec{f}(\mathbf{w}^{(1)} \vec{x}) - \mathbf{f}' \cdot \mathbf{w}^{(1)T} \vec{x} \right) \cdot |\vec{\zeta}| \right\} \\ & - \lambda_* \cdot \text{sign}(\vec{\zeta}) \cdot \text{sign}(\mu) \cdot \text{sign}(S_t) \cdot \|\vec{\zeta}\|_2 \cdot \left[\text{Spur} \left\{ \mathbf{w}^{(1)T} \mathbf{w}^{(1)} \right\} + \text{Spur} \left\{ \mathbf{w}^{(2)T} \mathbf{w}^{(2)} \right\} \right]. \end{aligned} \quad (\text{B.4})$$

Wie schon in Abschnitt 5.2.2 wird erneut die Vorzeichenkonvention der Lernrate $\text{sign}(\mu) = \text{sign}(S_t) \cdot \text{sign}(\vec{\zeta})$ eingesetzt, so dass alle sign-Terme verschwinden. Auf diese Weise wird ersichtlich, dass die Erweiterung der e-Modifikation um das Vorzeichen des gefilterten Fehlers letztendlich zu einer Form von \dot{V} führt, wie sie für den klassischen Backpropagation-Algorithmus unter anderem in [46] beschrieben wird. Unter Verwendung von $\mathbf{a}^T \mathbf{b} = \text{Spur}\{\mathbf{b} \cdot \mathbf{a}^T\}$ folgt somit:

$$\begin{aligned} \dot{V} = & \vec{\zeta} \cdot \mathbf{w}^{(2)T} \cdot \mathbf{f}' \cdot \mathbf{w}^{(1)T} \cdot \vec{x} + \vec{\zeta} \mathbf{w}^{(2)T} \cdot \left(\vec{f}(\mathbf{w}^{(1)} \vec{x}) - \mathbf{f}' \cdot \mathbf{w}^{(1)T} \vec{x} \right) \\ & - \lambda_* \cdot \|\vec{\zeta}\|_2 \cdot \left[\text{Spur}\{\mathbf{w}^{(1)T} \mathbf{w}^{(1)}\} + \text{Spur}\{\mathbf{w}^{(2)T} \mathbf{w}^{(2)}\} \right]. \end{aligned} \quad (\text{B.5})$$

In Gleichung B.5 kürzen sich nun die vom Netzeingangsvektor \vec{x} abhängigen Terme heraus, so dass sich die Zeitableitung der Kandidatenfunktion zu

$$\dot{V} = \vec{\zeta} \cdot \mathbf{w}^{(2)T} \cdot \vec{f}(\mathbf{w}^{(1)} \vec{x}) - \lambda_* \cdot \|\vec{\zeta}\|_2 \cdot \left[\text{Spur}\{\mathbf{w}^{(1)T} \mathbf{w}^{(1)}\} + \text{Spur}\{\mathbf{w}^{(2)T} \mathbf{w}^{(2)}\} \right] \quad (\text{B.6})$$

ergibt. Unter Verwendung der bereits eingeführten euklidischen Norm als Methode zur Berechnung einer Vektorbegrenzung kann eine obere Schranke aller Neuronenausgänge der verborgenen Schicht $\vec{f}(\mathbf{w}^{(1)} \vec{x})$ angegeben werden, da die verwendete Tangens Hyperbolicus-Funktion selber maximal den Wert 1 annehmen kann. Für die Neuronenanzahl $q^{(2)}$ und einem konstanten Biaswert von ebenfalls 1 folgt für die Vektornorm:

$$\left\| \vec{f}(\mathbf{w}^{(1)} \vec{x}) \right\|_2 = \sqrt{q^{(2)} + 1}. \quad (\text{B.7})$$

Mit Gleichung B.7 und der Frobeniusnorm der Netzwerkgewichte $\|\mathbf{w}\|_F$ ist es nach [46, 48] nun möglich, eine obere Schranke der abgeleiteten Ljapunow-Kandidatenfunktion anzugeben:

$$\dot{V} \leq \|\vec{\zeta}\|_2 \cdot \|\mathbf{w}\|_F \cdot \left(\sqrt{q^{(2)} + 1} - \lambda_* \cdot \|\mathbf{w}\|_F \right). \quad (\text{B.8})$$

Aus Gleichung B.8 folgt somit direkt, dass die Bedingung $\dot{V} \leq 0$ unabhängig von $\vec{\zeta}$ erfüllt wird, wenn

$$\|\mathbf{w}\|_F \geq \frac{\sqrt{q^{(2)} + 1}}{\lambda_*} \quad (\text{B.9})$$

gilt. Damit handelt es sich um die identische obere Gewichtsgrenze, wie sie auch für den Fall des klassischen Backpropagation-Algorithmus in [46] hergeleitet wird. Es kann somit gezeigt werden, dass auch für die Nutzung eines Gleitzustandstrainings und einem entsprechend angepassten Lerngesetz die Gewichtsmatrix \mathbf{w} nach oben begrenzt ist.

Anhang C

Weitere Ergebnisse

C.1 Flugsimulation

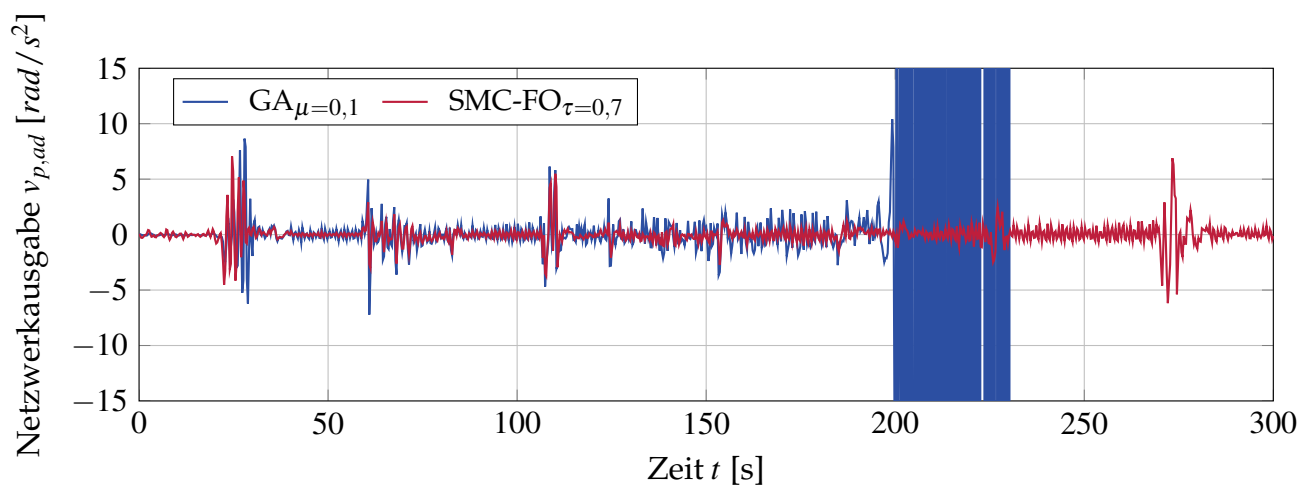


Abbildung C.1: Vergleich der Netzwerkausgabe für die Rollratenregelung mit Backpropagation- und Gleitzustandstraining im Schadensfall

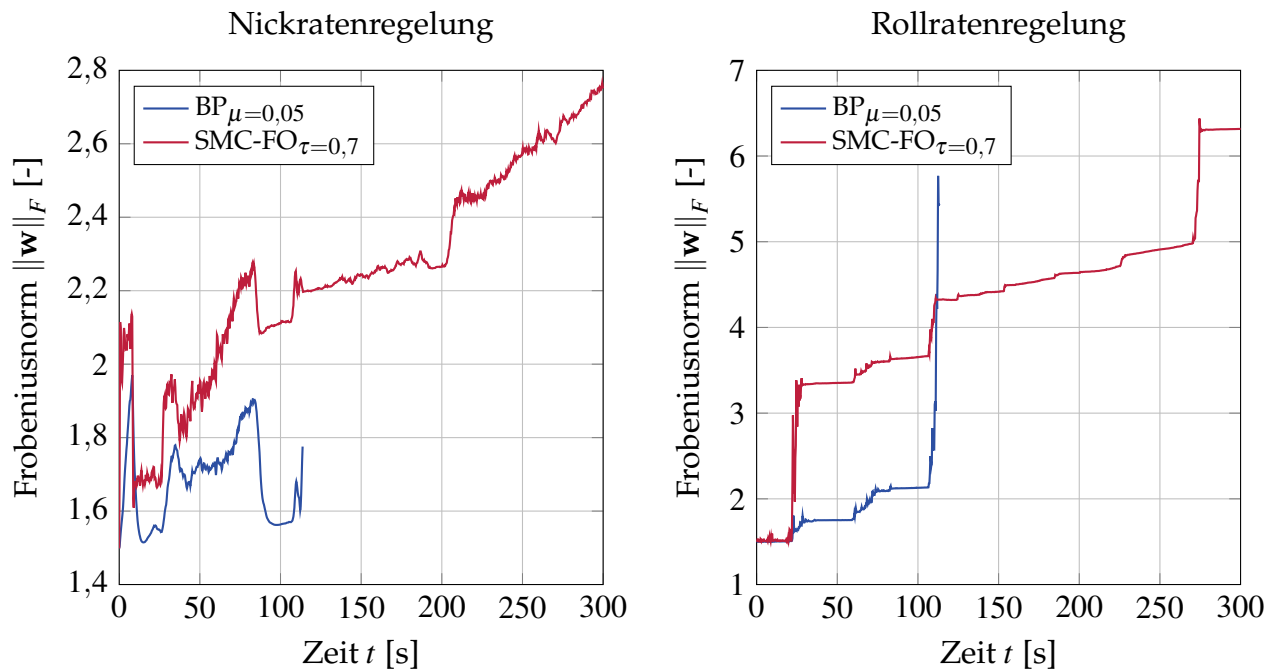


Abbildung C.2: Vergleich der Frobeniusnorm für ein Backpropagation- und Gleitzeittraining im Schadensfall mit zusätzlich induziertem Inversionsfehler

C.2 Fahrsimulation

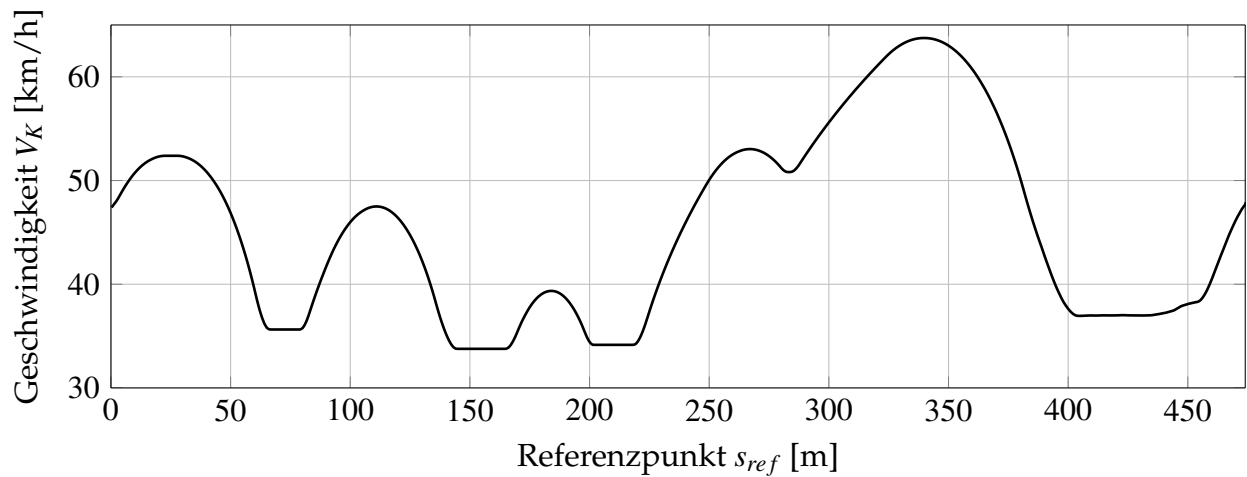


Abbildung C.3: Geschwindigkeitsvorgabe für die Langzeituntersuchung auf Rundkurs

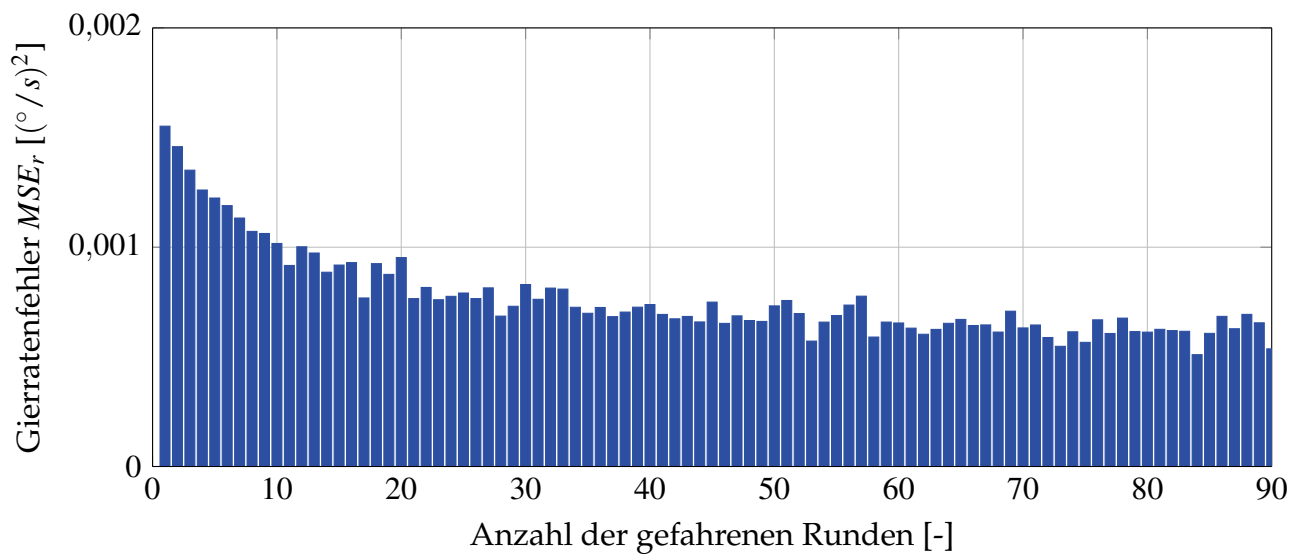


Abbildung C.4: Mittlerer quadratischer Fehler der Gierratenregelung über 90 Runden mit SMC-FO

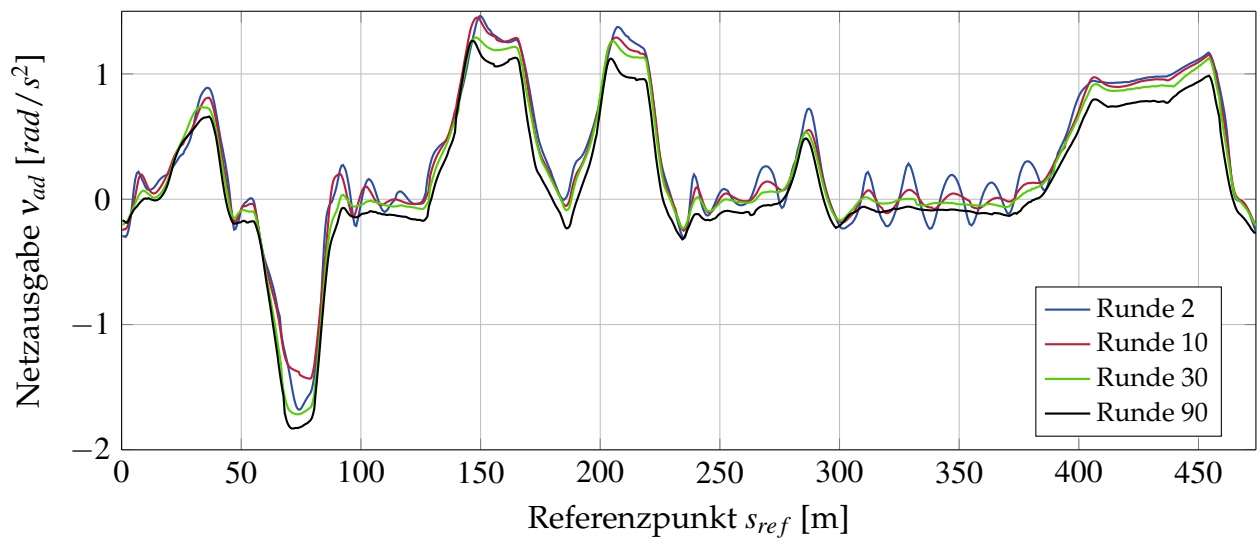


Abbildung C.5: Vergleich der Netzwerkausgabe für mehrere Runden mit SMC-FO

C.3 Fahrversuch

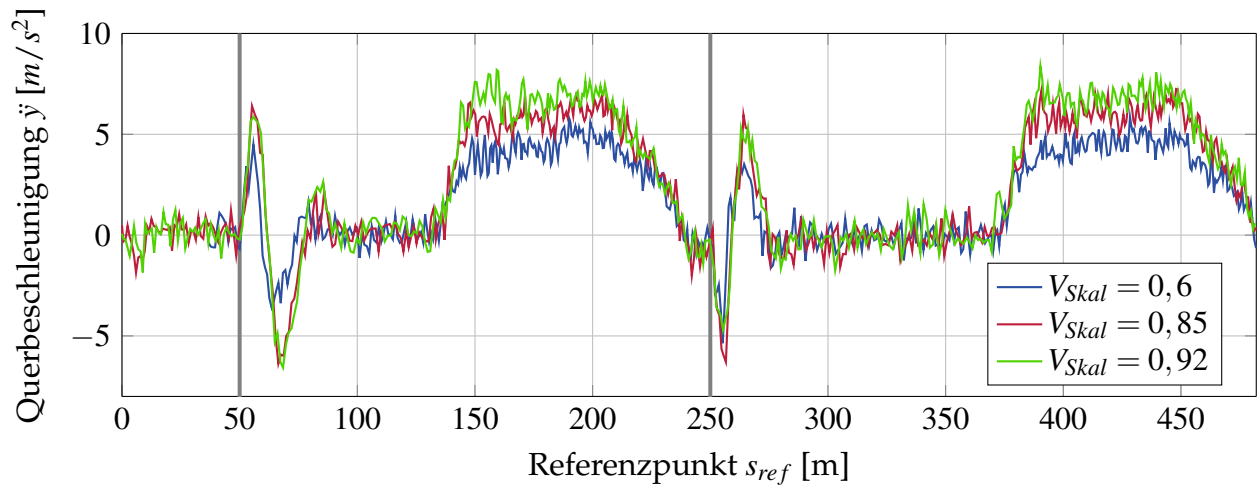


Abbildung C.6: Vergleich der Querschleunigungen für die unterschiedlichen Geschwindigkeitsvorgaben mit Fehleraufschaltung

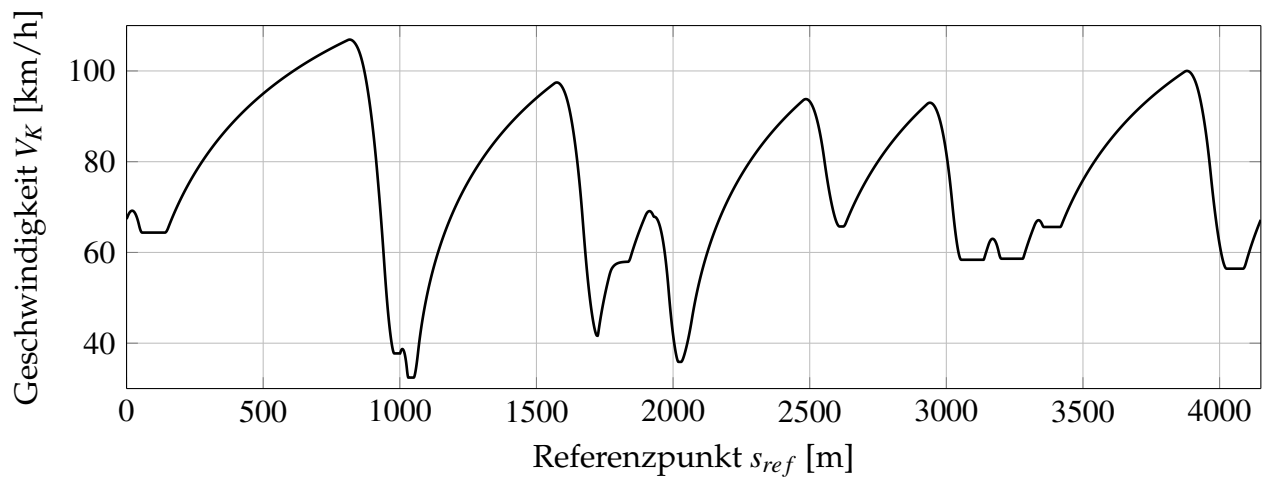


Abbildung C.7: Geschwindigkeitsvorgabe für den Fehlerausgleich auf anspruchsvollem Rundkurs